

# Simulation study of a remote wireless path tracking control with delay estimation for an autonomous guided vehicle

Camilo Lozoya · Pau Martí · Manel Velasco ·  
Josep M. Fuertes · Enric Xavier Martín

Received: 9 February 2010 / Accepted: 12 May 2010 / Published online: 5 June 2010  
© Springer-Verlag London Limited 2010

**Abstract** Industrial applications involving mobile systems can benefit from the use of wireless technology. However, wireless communication has not been widely accepted on the factory floor due to its difficulty in achieving the timely and reliable transmission of messages over error-prone wireless channels. This paper presents an autonomous guided vehicle (AGV) path tracking wireless control system where an accurate delay estimation scheme is shown to be the key for successful operation. The control architecture consists on an AGV connected through a wireless network to a controller. To mitigate the negative effects that varying time delays in data transfer have in the networked control loop, the controller performs two tasks at each loop operation. First, it uses a Kalman filter to produce an optimal delay estimate considering a simple stochastic

model of the wireless delay dynamics. Second, each delay estimate is employed to infer the real AGV position which permits to compute the appropriate control commands. Results show that the proposed technique provides more efficient and effective operation for path tracking control compared to similar previously proposed solutions.

**Keywords** Autonomous guided vehicle · Remote path tracking · Time delay estimation

## 1 Introduction

The deployment of wireless technologies in the industrial environment plays an important role in the simplification of costly and complex infrastructures and also in the improvement of flexibility in the factory. The localization and tracking of components, the coordination of autonomous transport vehicles and mobile robots, the monitorization and control in hazardous and difficult-to-access environments, and applications involving distributed control are all areas in which wireless technologies could be used in an industrial environment [1].

Industrial applications involving mobile systems can benefit from the use of wireless communication technologies [2]. Usually, mobile systems such as AGVs and mobile robots contain an on-board control system that locally performs all control activities (see [3] and references therein for recent advances on local control for mobile systems). An alternate architecture involves remote control [4, 5]. In this new scheme, a remote controller performs some of the tasks previously performed in the vehicle on-board system. This schema

---

This work was partially supported by the ArtistDesing Network of Excellence on Design for Embedded Systems, IST-2008-214373, and by the Spanish project C3DE, CICYT DPI2007-61527.

---

C. Lozoya · P. Martí (✉) · M. Velasco · J. M. Fuertes ·  
E. X. Martín  
Automatic Control Department, Technical University  
of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain  
e-mail: pau.marti@upc.edu

C. Lozoya  
e-mail: camilo.lozoya@upc.edu

M. Velasco  
e-mail: manel.velasco@upc.edu

J. M. Fuertes  
e-mail: josep.m.fuertes@upc.edu

E. X. Martín  
e-mail: enric.xavier.martin@upc.edu

opens up a full set of new applications based on remote and decentralized control which is expected to lead to higher flexible manufacturing systems. Based on the benefits provided by the remote control, an adequate architecture for AGV systems may consist on an external controller sending and receiving the control commands to the mobile vehicle through a wireless network.

For control applications requiring closing the loop over a communication network, i.e., networked control systems [6], the network must ensure that the quality of communication is comparable with the quality provided by the point-to-point connections between controllers and sensors/actuators. When the communication infrastructure does not fulfil this demand, the overall system performance degrades. Hence, new solutions are required that can either focus on designing robust and loss-tolerant applications and control algorithms [7], on trying to improve the communication reliability and quality [8, 9], or both [10, 11].

When communication is served by wired fieldbus systems, the stringent reliability and timeliness requirements are met [12], that is, guarantees can be given on the eventual delivery of packets and delivery times, respectively. When wireless technologies are applied instead, the same type of transmission quality of traditional wired technologies should be achieved. However, wireless technologies are not as good as wired ones in achieving the timely and reliable transmission of messages due to particular characteristics of the radio channels such as error properties, signal propagation, mobility aspects, etc., as well as possibly non-deterministic medium access control mechanisms. This poses new challenges in the design of wireless control applications [13] and demands solutions where the characteristics of the radio channels and medium access control must be taken in consideration and managed properly.

The lack of timeliness and reliability of the wireless communication system produces severe varying time delays for packet transmissions [1]. This paper presents a novel approach to remote wireless AGV path tracking control that accounts for the varying time delays caused by the wireless network. The AGV path tracking system consists on the autonomous vehicle connected to the controller through a wireless network. Given the position of the vehicle sent by the AGV and taking into account the set-point, the controller uses a generalization of the quadratic curve approach [14] for path tracking control to compute and send the control commands to the AGV. As outlined in [6] and further investigated in [4, 5], remote path tracking control schemes demand accurate delay estimation techniques

for successful operation of the networked control loops. Otherwise, severe performance degradation may occur.

Rather than designing specific controllers capable of dealing with systems with varying time delays, in order to mitigate the negative effects of the varying time delays, the presented controller performs two tasks at each closed loop operation. First, it uses a Kalman filter [15] to produce an optimal delay estimate considering a simple stochastic model of the wireless delay dynamics [16]. Second, each delay estimate is employed to infer the real AGV position which permits to compute the appropriated control commands.

A simulation model has been implemented in order to evaluate the effectiveness of the proposed approach by measuring the vehicle's path deviation and the total traveling time for different paths and network traffic conditions. Results show that the proposed approach provides the most efficient and effective operation for wireless path tracking control compared to similar previously proposed schemes that rely on delay estimation techniques such as [17] and [18]. In addition, results show that the presented approach is robust with respect to different network load scenarios. This work consolidates and extends previous results [19].

The paper contributions can be summarized as follows: The first contribution is to present a novel wireless path tracking control algorithm that incorporates an optimal delay estimator that permits to infer, at the remote controller side, the correct AGV position for computing each control signal. The second contribution is to show that the accuracy of the delay estimator embedded in the control algorithm outperforms previous techniques used for network delay estimation.

The rest of the paper is organized as follows: Section 2 describes the wireless network model as well as the AGV model. Section 3 presents the remote path tracking approach and the Kalman filter delay estimator. Section 4 describes the simulation setup, and Section 5 summarizes the key results. Finally, Section 6 concludes the paper.

## 2 Preliminaries

The networked control system consists on an AGV connected to the controller through a wireless network. This section establishes the network delay and the AGV models.

### 2.1 Network delay metric and model

Commercially available and widespread wireless networks include Bluetooth (IEEE 802.15.1) [20], ZigBee

(IEEE 802.15.4) [21], and IEEE 802.11 Wireless Local Area Network technology [22–24], the later commonly known as WiFi. Since Bluetooth and ZigBee are short range solutions, the study carried out in this paper focuses on WiFi. WiFi was originally conceived as the wireless extension of Ethernet and now has become one of the most mature wireless communication systems [25]. The WiFi standard IEEE 802.11 includes several specifications, namely 802.11 a/b/g and e, with differences at the physical layer but with a common medium access control protocol based on the carrier sense multiple access (CSMA) mechanism. The basic access requires nodes to sense the channel idle for a specific time interval before attempting any packet. In case the channel is sensed busy, nodes defer their transmission attempts to a later time, on the basis of a backoff mechanism. Finally, packet transmission have to be acknowledged by the received to the transmitter.

High-quality data transfer is difficult to meet in WiFi networks because of the adverse properties of the radio channels and its medium access control. Error prone radio channels demand different sorts of forward error correction techniques as well as automatic repeat request protocols. The CSMA mechanism introduces random and long access delays when there is a heavy traffic load. Both end up producing long varying time delays for packet transmissions.

An important metric for assessing the quality of wireless communication link is the network delay [1]. One of the most common metrics in networks modeling is the round trip time delay, RTT [26, 27]. Typically, the RTT delay is the time elapsed from sending a message to the receiver plus the time required for the receiver to generate and send back an answer, up to the time it gets back to the sender.

For the remote path tracking scheme presented in this paper, the RTT delay is the time spent from the AGV sending the current position to the controller up to the time when the AGV receives the control command, i.e., delay from sampling to actuation. Considering the transmission time at both sides (AGV and controller) constant, the control command computation time constant, and the varying propagation delays (varying due to the mobility of the AGV) contributing with a bias, the major contribution to the RTT delay is given by the link delay.

Due to the high bandwidth fluctuation of WiFi links, for a dynamic system modeling point of view, a single RTT sample is useless in the sense that it does not provide much information about what RTT will experience the next transmission. Instead of considering the instantaneous RTT, a mean value would be a better approach. In addition, in an industrial setting with a

known and fixed number of AGVs sending and receiving control data over a private WiFi, communication delays, in average, can be considered constant. Therefore, we regard the RTT delay as a smooth signal and the high frequency component of it as a noise disturbance. Hence, we can model the signal as a standard example of noisy observations of a constant [28]

$$\tau_{(k+1)} = \tau_{(k)} + w_{(k)} \tag{1}$$

$$y_{(k)} = \tau_{(k)} + v_{(k)} \tag{2}$$

where  $\tau_{(k)}$  represents the RTT delay,  $w_{(k)}$  is a zero mean, Gaussian white process noise with covariance  $Q$  that models the bias introduced by propagation delays, and  $v_{(k)}$  is a zero mean, Gaussian white measurement noise with covariance  $R$  that models the noisy link characteristics of the RTT.

### 2.2 AGV description and model

The AGV we consider consists in a differential drive mobile robot with two driving wheels and one caster wheel. As shown in Fig. 1, the AGV current position is given by  $x_C, y_C$  and the current azimuth angle is  $\theta_C$ .

In the AGV kinematic model, the vehicle states can be calculated from the rotational velocities of the right and left wheels, defined by  $\omega_r$  and  $\omega_l$ ,

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{W} & -\frac{r}{W} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \tag{3}$$

where  $v$  is translational velocity,  $\omega$  is the rotational velocity,  $r$  is the wheel radius, and  $W$  represents the

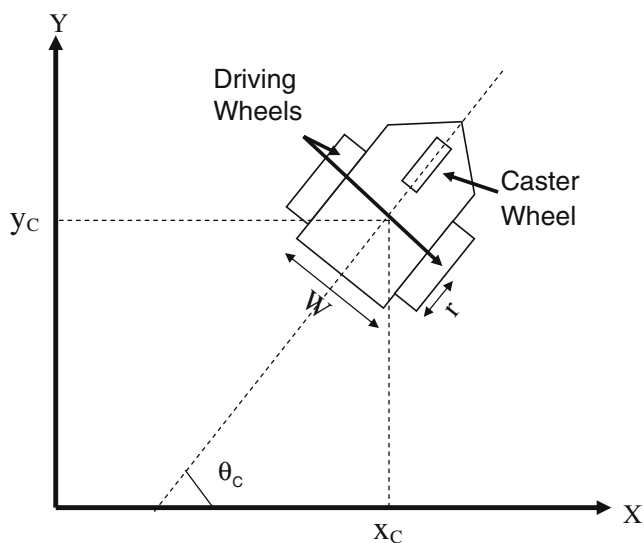


Fig. 1 Schematic of the AGV

distance between the two wheels of the vehicle. The AGV next position and attitude, after a  $\Delta t$  time interval has elapsed, is obtained from the translational and rotational velocities.

$$x_{C(k+1)} = x_{C(k)} + v\Delta t \cos(\theta_{C(k)} + \omega \frac{\Delta t}{2}) \quad (4)$$

$$y_{C(k+1)} = y_{C(k)} + v\Delta t \sin(\theta_{C(k)} + \omega \frac{\Delta t}{2}) \quad (5)$$

$$\theta_{C(k+1)} = \theta_{C(k)} + \omega\Delta t \quad (6)$$

where  $\Delta t$  represents the sampling time interval.

### 2.3 Path tracking control algorithm

The goal of the path tracking control is to drive the vehicle to a desired location or reference point by controlling the speed of the vehicle's right and left wheels in such a way that the deviation from the shortest path is minimized. The quadratic curve approach proposed in [14] is used as a path tracking algorithm. The main goal of this algorithm is to move the AGV along a quadratic curve to a reference point on a desired path. The advantage of this algorithm is that only AGV's position must be controlled because it is assumed that the orientation of the AGV moves close to the desired value in the motion along the reference path. Hence, the algorithm is suitable for delay-sensitive applications because of its simple computation with minimal amount of information.

The path tracking control algorithm would essentially do the following: First, it would read the vehicle current position  $[x_C, y_C, \theta_C]$  and the reference point  $[x_R, y_R, \theta_R]$ . Then, it would process them to compute the error vector in the AGV coordinate as

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_C & \sin \theta_C & 0 \\ -\sin \theta_C & \cos \theta_C & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R - x_C \\ y_R - y_C \\ \theta_R - \theta_C \end{bmatrix}. \quad (7)$$

By using the error vector, the quadratic curve in the robot coordinate can be calculated as

$$y = A_{(k)}x^2 \quad \text{where} \quad A_{(k)} = \text{sign}(e_x) \frac{e_y}{e_x^2}. \quad (8)$$

Considering that the AGV runs forward when  $e_x > 0$ , the constraint equations for the AGV to move along the calculated quadratic curve are

$$v_{(k)} = \text{sign}(e_x) \sqrt{\dot{x}^2 (1 + 4A_{(k)}^2 x^2)} \quad (9)$$

$$\omega_{(k)} = \frac{2A_{(k)}\dot{x}^3}{v_{(k)}^2}. \quad (10)$$

If  $x$  at the time instant  $k\Delta t \leq t < (k+1)\Delta t$  is given by

$$x = K_{(k)}(t - k\Delta t) \quad \text{and} \quad K_{(k)} = \text{sign}(e_x) \frac{\alpha}{1 + |A_{(k)}|} \quad (11)$$

where  $\alpha$  is a positive constant used as a speed factor, then the desired linear and angular velocities in Eqs. 9 and 10 can be approximated to

$$\hat{v}_{R(k)} \simeq K_{(k)} \quad (12)$$

$$\hat{\omega}_{R(k)} \simeq 2A_{(k)}K_{(k)}. \quad (13)$$

The desired inputs 12 and 13 to the AGV are used to calculate the desired angular velocities for the right and left wheels using Eq. 3

$$\omega_{R,r} = \frac{\hat{v}_{R(k)}}{r} + \frac{W\hat{\omega}_{R(k)}}{2r} \quad (14)$$

$$\omega_{R,l} = \frac{\hat{v}_{R(k)}}{r} - \frac{W\hat{\omega}_{R(k)}}{2r}. \quad (15)$$

These speeds, to be applied to the AGV, will be controlled in the AGV side using a traditional PI controller.

## 3 Remote AGV path tracking control

### 3.1 Problem to be solved

In the remote path tracking control scheme, it has to be considered that delays may provoke an inconsistent operation of the previous control algorithm. The accuracy of the path tracking control algorithm summarized above relies on two assumptions. First, at each control algorithm execution, the AGV current position must be known. And second, the time elapsed from reading the AGV current position (sampling) to notifying the AGV with the desired right and left wheels angular velocities (actuation) is negligible.

However, in the wireless path tracking control setting, where the remote controller receives the AGV position and computes the wheels angular velocities, none of the two assumptions is met. First, the AGV position received by the remote controller is no longer the current position of the AGV at the time this information is received in the remote controller due to the transmission delay from the AGV to remote controller. Hence, control signals would be computed using inaccurate data. Second, the delay from sampling to actuation is not negligible due to the RTT delay. Hence, control signals cannot be applied when demanded, which produces an accumulated error with respect to the desired trajectory.

The algorithm presented next tackles the first problem. The intuitive idea is that, in the remote controller, the received AGV position has to be adjusted to account for the wireless network delay. This is achieved by inserting in the remote control structure a network delay estimator that produces at each control algorithm computation the delay estimate that is used to infer the current position of the AGV from the position sent by the AGV.

### 3.2 Remote control algorithm

The proposed AGV path tracking networked control system is based on the scheme defined in [4] and [5] which is a modular structure and can be implemented over an existing control system. As a difference, the proposed approach includes a delay estimation algorithm based on the Kalman filter. The sequence of activities executed at each sampling period in the networked control system, Fig. 2, is the following:

- In the AGV.
  1. Obtain vehicle current position  $(x_C, y_C, \theta_C)$ .
  2. Calculate the previous RTT delay value  $\tau_{(k-1)}$ .
  3. Send both values to the remote controller.
- In the controller, after receiving the data sent by the vehicle.
  1. The route planner provides the next reference point  $(x_R, y_R, \theta_R)$ , if the previous one has been already reached.
  2. The network delay estimator (NDE) takes the previous RTT delay value  $\tau_{(k-1)}$ .
  3. The NDE estimates the next value of the RTT delay  $\hat{\tau}_{(k)}$ .
  4. The position predictor (PP) uses the estimated RTT delay value  $\hat{\tau}_{(k)}$  and the current position of the vehicle  $(x_C, y_C, \theta_C)$ . Since the PP knows

the kinematic behavior of the AGV, according with Eqs. 3–6, it calculates the estimated position of the vehicle  $(\hat{x}_C, \hat{y}_C, \hat{\theta}_C)$ .

5. The PP sends the estimated position to the internal controller.
6. The internal controller generates and sends the control signals  $(\omega_{R,r}, \omega_{R,l})$  to the AGV.

– In the AGV.

1. Update wheels velocity upon reception of the control signals.
2. Wait for the next sampling period.

The remote AGV path tracking control requires an accurate estimation of the RTT delay. It is important to notice that the AGV receives the control information under the assumption that the vehicle is in a specific location at time  $t + \hat{\tau}_{(k)}$ , but the control data are actually received at time  $t + \tau_{(k)}$ . If the difference between the estimated RTT delay  $\hat{\tau}_{(k)}$  and the actual RTT delay  $\tau_{(k)}$  is high, then the performance degradation of the AGV will increase.

### 3.3 Network delay estimation

Statistical algorithms (mean value, median value, max value), algorithms based on Markov formalisms, and exponential averaging algorithms to estimate RTT delay have been already proposed in [17] and [18]. However, in this paper, we propose to use a Kalman filter estimator because, considering the RTT delay model 1–2 and under the assumption of Gaussian noise components, it produces optimal linear estimates.

The Kalman filter estimates a process by using a form of feedback control. The filter estimates the process state at some time and then obtains feedback in the form of noisy measurements that are used to adjust the estimates. The equations for the Kalman filter are divided in two groups: time update equations (predictor) and measurement update equations (corrector).

The time update equations, Eqs. 16 and 17, are responsible for obtaining the *a priori* estimates. For the remote path tracking control algorithm, the estimated RTT delay  $\hat{\tau}_{(k)}$  corresponds to the Kalman filter's *a priori* estimation given by

$$\hat{\tau}_{(k)}^- = \hat{\tau}_{(k-1)} \tag{16}$$

where  $\hat{\tau}_{(k-1)}$  is the previous *a posteriori* estimate of the RTT delay. The *a priori* estimation of the covariance error is

$$P_{(k)}^- = P_{(k-1)} + Q \tag{17}$$

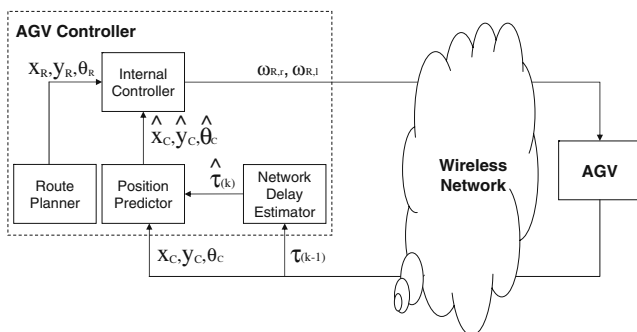


Fig. 2 Data-flow of the path tracking control over a network

where  $P_{(k-1)}$  is the previous a posteriori estimate of the covariance error and  $Q$  is the covariance process noise.

The measurement update equations, Eqs. 18, 19, and 20, are responsible for obtaining an improved a posteriori estimate. The feedback gain to be used in the next state and covariance estimation is computed as follows:

$$K_{(k)} = \frac{P_{(k)}^-}{P_{(k)}^- + R} \quad (18)$$

where  $K_{(k)}$  is the Kalman gain,  $H$  defines the constant measurement gain,  $R$  is the covariance value of the measurement noise. In order to improve the accuracy of the a posteriori estimate of the current state, the a priori estimate is adjusted using a new measurement, as follows:

$$\hat{\tau}_{(k)} = \hat{\tau}_{(k)}^- + K_{(k)}(y_{(k)} + \hat{\tau}_{(k)}^-) \quad (19)$$

where  $y_{(k)}$  is the measured output of the system 2. The next covariance estimation is given by

$$P_{(k)} = P_{(k)}^- (1 - K_{(k)}) \quad (20)$$

where  $P_{(k)}$  it the a posteriori estimation of the covariance error.

## 4 Simulation setup

To assess the proposed remote path tracking algorithm including the effectiveness of the Kalman-based RTT delay estimation technique, extensive simulations have been performed. Reported results cover different algorithms for remote path tracking control, evaluated for several prototype routes and under different WiFi network traffic conditions.

### 4.1 Other estimation algorithms

This section summarizes previously proposed RTT delay estimation algorithms [17] and [18] that will be used in the evaluation of the solution presented in this paper. These algorithms can be divided into:

- Statistical estimation algorithms: the mean (Eq. 21), median (Eq. 22), and max (Eq. 23) value estimation algorithms predict the current RTT delay by applying statistical functions to the last  $w$  measured RTT delays (Eq. 21), where  $w$  is known as the window

size and its value defines the amount of previous data that needs to be stored.

$$\hat{\tau}_{k+1|w} = \text{mean}\{\tau_k, \tau_{k-1}, \dots, \tau_{k-w+1}\} \quad (21)$$

$$\hat{\tau}_{k+1|w} = \text{median}\{\tau_k, \tau_{k-1}, \dots, \tau_{k-w+1}\} \quad (22)$$

$$\hat{\tau}_{k+1|w} = \max\{\text{mean}(\tau_k, \tau_{k-1}, \dots, \tau_{k-w+1}), \text{median}(\tau_k, \tau_{k-1}, \dots, \tau_{k-w+1})\} \quad (23)$$

- Markov chain estimation algorithm: given the current RTT delay state and taking previously defined probability distributions, the new estimate is obtained. In particular, it predicts the delay range by classifying the Markov states and its state-transition matrix. The Markov states at time  $k$  are denoted as  $\{1, \dots, S\}$ , where  $S$  is the total number of states. The state-transition matrix, denoted as  $R_{S \times S}$ , captures the dynamic changes from state to state, where its values are calculated from the probability distribution of a previously defined dynamic state changes. Therefore, the estimated time delay,  $\hat{\tau}_k$ , is represented by current state of the Markov matrix, and the next estimated value,  $\hat{\tau}_{k+1}$ , is obtained when the Markov chain moves to the next state using the state-transition matrix.
- Exponential averaging estimation algorithm: It smoothes and averages a sequence of values along a time series. It makes a prediction based on a smoothed series  $P_{k-1}$  as well as the current RTT delay

$$P_k = \alpha \tau_k + (1 - \alpha) P_{k-1} \quad (24)$$

$$\hat{\tau}_{k+1} = P_k. \quad (25)$$

This algorithm corresponds to the steady-state Kalman filter if  $\alpha$  was the Kalman gain.

### 4.2 Generation of delays and network load

Varying RTT delays of a WiFi network present an unimodal and asymmetric distribution with a long tail in the right side. Therefore, a Gamma distribution can be used to represent the behavior of the end-to-end delay in WiFi networks [27]. In addition, experimental data reveal that, irrespective of the traffic crossing the link, the empirical histograms almost have the same general shape, which can be fitted by a Gamma distribution with different parameters for almost 90% of the distributions [29].

According to these indications, the simulation setup randomly generates the current RTT delay values  $\tau(k)$  based on different gamma distributions functions de-

pending on the network load. Different network loads were considered during simulation. Each particular load is characterized by an specific average RTT delay value:

- High delay variability load, with RTT delay mean of 35 ms, shortest delay of 5 ms, and longest delay of 95 ms
- Low delay variability load, with RTT delay mean of 35 ms, shortest delay of 25 ms, and longest delay of 55 ms
- Low delay variability with different RTT delay means, which correspond to six loads. The first load has RTT delay mean of 10 ms, the second of 20 ms, etc., up to 60 ms. The distribution with mean of 60 ms is the last one in order to avoid having delays larger than 95 ms

It is important to point out that the period of the control algorithm is 100 ms.

#### 4.3 Algorithm parameters

For the simulations, the key parameters of each delay estimation technique have been empirically tuned to obtain the best performance possible for remote path tracking control in the case of network load with high RTT delay variability.

For the Kalman filter algorithm, the noise covariance  $Q$  of the RTT delay process has a small but non-zero value ( $Q = 0.0001$ ) in order to model the bias introduced by propagation times of the data transfer to a mobile vehicle. The noise covariance for the measurements modeling the high-frequency components of the measurement noise  $R$  is greater than one. In particular, for the high RTT delay variability distribution, given that its standard deviation is 15.26 ms, the measurement variance has been fixed at  $R = (15.26)^2 = 233$ . The initial value of  $P_{(k-1)}$  is different from zero ( $P_{(k-1)} = 1$ ). For the statistical algorithms, experiments have shown that the longer the window size, the better the performance of the remote tracking. However, beyond a window size of 10, no increment in performance was noticed. Therefore,  $w$  was set to 10. The control parameter for the exponential averaging has been specified  $\alpha = 0.1$  while the first value for  $P_{k-1}$  is set to the network delay average, 35 ms. The transition matrix probabilities for the Markov chain algorithm has been obtained from the density values taken from the gamma distribution in such a way the remote path tracking performance is maximized.

It is important to notice that though the actual network load may vary, the algorithms parameters, once defined, are not changed during simulations.

#### 4.4 Simulation scenarios

In addition to the remote path tracking control where delay estimation is carried out by the proposed solution or using the explained existing techniques, for the comparative analysis, the following scenarios have been also considered:

- No delay. There is no delay in the wireless network  $\tau_{(k)} = 0$ , so the AGV receives instantaneously the control information. It represents the behavior of the model considering an ideal communication environment or a local controller.
- Ideal estimation. The estimated RTT values equal to the current RTT delay,  $\hat{\tau}_{(k)} = \tau_{(k)}$ . This scenario includes the adverse effects of the delay, but the effects are minimized by an ideal estimation.
- No estimation. The controller does not make any delay compensation. This represents the worst case scenario since the delay effects are maximized.

#### 4.5 Cost functions

The goal of path tracking control is to accurately follow trajectories. Two cost functions are used to evaluate the performance of the proposed algorithm. The first cost function,  $J_1$ , represents the quadratic deviation of the AGV from the reference points,

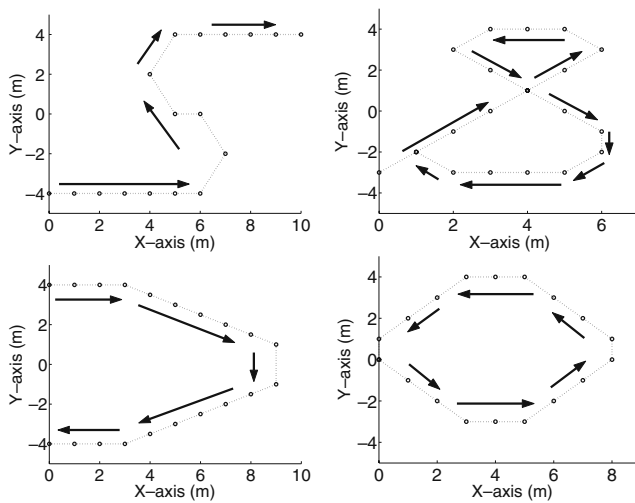
$$J_1 = \sum_{k=t_0}^{t_f} D(p_{c(k)}, p_{r(k)})^2 \quad (26)$$

where  $D$  is the distance between the AGV's current position  $p_{c(k)}$  and the next reference point  $p_{r(k)}$  defined by the route planner,  $t_0$  is the initial time (for the proposed model  $t_0 = 0$ ), and  $t_f$  is the final time when the AGV completes the route. The second cost function,  $J_2$ , is the total traveling time spent by the AGV to reach the final destination.

$$J_2 = t_f - t_0 \quad (27)$$

#### 4.6 Routes

Four different routes were defined to test the proposed algorithm. They capture key aspects such as different types of curves, straight lines of different longitude, etc. in order to cover an extended topology, as it is illustrated in Fig. 3.



**Fig. 3** Different routes to evaluate AGV movements

## 5 Simulation results

First, the remote path tracking algorithm using the Kalman filter estimation technique is shown to perform effective tracking of a given route regardless of its operation in a networking environment that presents high RTT delay variability load. Second, the proposed algorithm is shown to provide robust tracking of a given route if the network delay assumptions (high RTT delay variability) do not match the actual delays (low RTT delay variability). And finally, the last set of simulations show that the proposed algorithm is the best among the studied ones regardless of changes in the network load.

### 5.1 Effectiveness of the remote path tracking approach

The first evaluation focuses on the effectiveness and efficiency of the proposed algorithm. To this extent, the algorithm using the Kalman estimator is compared to the case where no estimation is performed for the first prototype route and taking into account the worst delay scenario, that is, the high RTT delay variability. This serves two purposes. First, it looks into the negative impact that the RTT delay has in the remote path tracking control (no estimation scenario), and second, it highlights the efficiency of the solution presented in this paper.

The adverse effects of the RTT delay and the effectiveness of the presented algorithm affect both the deviation of the vehicle from the given route and the traveling time. In the case of no delay, the vehicle quadratic deviation from the original path is  $29.72\text{m}^2$ . When delays enter in the closed loop and no estimation process is available the deviation goes up to  $61.32\text{m}^2$

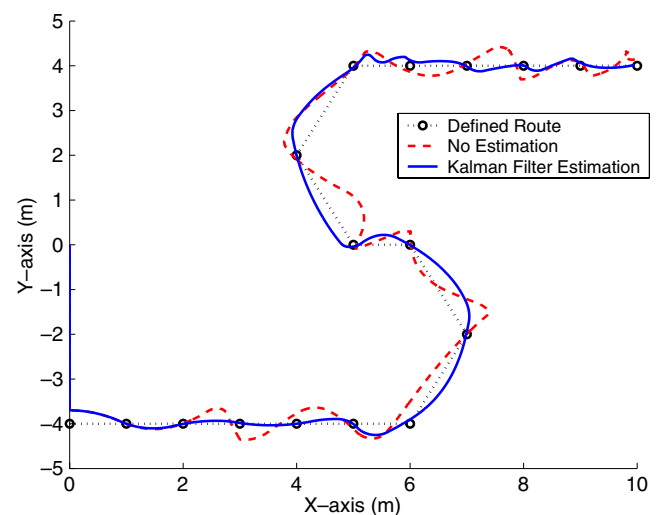
while the proposed approach achieves a deviation of  $42.84\text{m}^2$ . The later approaches the case of ideal estimation,  $41.74\text{m}^2$ . The traveling time evaluation presents similar numbers. For the case of no delay, the total tracking time is 50.00 s. In the presence of delays, the presented approach halves the increase in time of the no estimation case. This evaluation corroborates that estimating the delay in order to compute the control commands using accurate data becomes crucial for remote path tracking control.

Figure 4 illustrates the trajectories followed by the AGV for the case of no estimation and for the presented algorithm. The trajectory followed by the vehicle presents dramatic deviations with respect to the defined route for the no estimation scenario. However, the figure clearly shows that the proposed solution using the Kalman filter mitigates the negative effects of the delays in the networked control system, providing a more accurate tracking.

For performance analysis purposes, Table 1 summarizes the performance numbers of the remote path tracking algorithm using the other estimation techniques (Section 4.1) for the two cost functions. As it can be clearly seen, among the other estimation methods, remote path tracking using the Markov-based estimation and the exponential averaging estimation provides the next best performance after the Kalman filter one.

### 5.2 Robustness in front of unexpected delays

The previous evaluation of the remote path tracking control algorithm was performed considering that the



**Fig. 4** Vehicle's trajectories resulting of the remote path tracking control when, in the presence of delays, no delay estimation occurs and when Kalman filter estimation is applied

**Table 1** Remote path tracking control performance evaluation of all tested estimation methods for the first route under high RTT delay variability

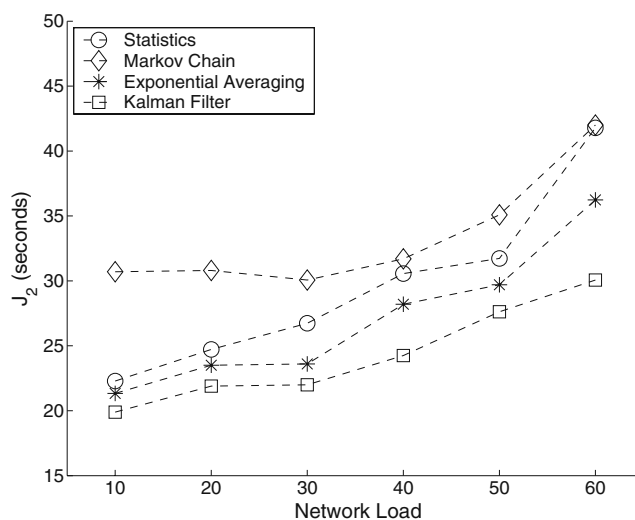
Simulation scenario	$J_1$ (m <sup>2</sup> )	$J_2$ (s)
No delay	29.72	50.00
Ideal estimation	41.74	54.50
No estimation	61.32	72.50
Statistical estimation: mean value	49.64	70.70
Statistical estimation: median value	48.29	69.80
Statistical estimation: max value	50.36	70.50
Markov chain estimation	45.28	65.10
Exponential averaging estimation	46.58	64.10
Kalman filter estimation	42.84	61.40

actual WiFi delays coincide with those used in the tuning of the control parameters of all the evaluated estimation techniques. Table 2 shows the final values of  $J_1$  and  $J_2$  for the case of having actual RTT delays with lower variability than those used in the control design, for the first route. The first observation is that in general these performance numbers are better when compared to the evaluation reported in Table 1. The reason for this is that the actual low RTT delay variability presents shorter delays compared to the high RTT delay variability, and therefore, deviations and traveling time are shorter. However, the accuracy of the remote path tracking control using the Kalman estimation reflected for example in the deviation metric for the performance reported in Table 1 is of 97% while for the one of Table 2 is of 93%. This is because the control parameters were tuned for the delay distribution of the high RTT delay variability.

Apart from this consideration, Table 2 shows that among the estimation algorithms the remote path tracking control using the Kalman filter presents the best performance when considering the path deviation and the traveling time. In addition, Table 2 reveals an interesting property. The control algorithm using

**Table 2** Remote path tracking control performance evaluation of all tested estimation methods for the first route with unexpected RTT delay variability

Simulation scenario	$J_1$ (m <sup>2</sup> )	$J_2$ (s)
No delay	29.72	50.00
Ideal estimation	39.07	51.20
No estimation	56.80	67.70
Statistical estimation: mean value	45.07	59.30
Statistical estimation: median value	45.00	58.40
Statistical estimation: max value	45.72	59.00
Markov chain est.	54.20	66.20
Exponential averaging estimation	43.72	57.20
Kalman filter estimation	41.94	54.20

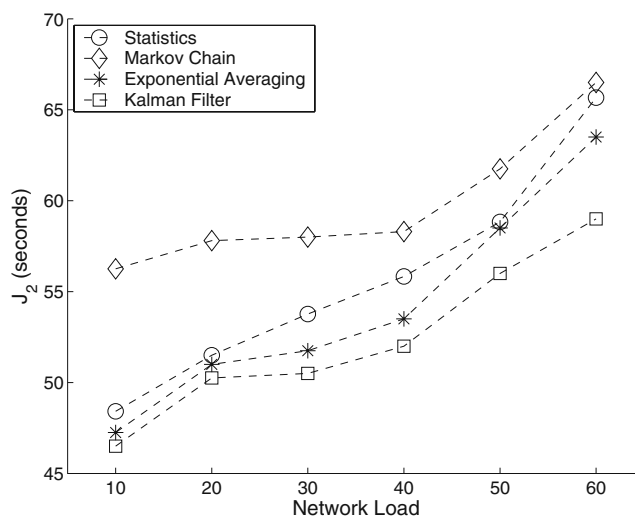


**Fig. 5** Tracking deviation ( $J_1$ ) results for different network loads

the Kalman filter is robust against unexpected delays compared to the Markov-based algorithm. The later performs much worse because the transition probability matrix used in the design does not hold anymore if the actual delays do not match the assumed ones. And although control algorithms using statistical estimation methods or using the exponential averaging method are also robust against unexpected delays, the approach with Kalman still performs best.

5.3 Robustness in front of different loads

The last performance analysis involves evaluating the remote path tracking control algorithm for the six low



**Fig. 6** Traveling time ( $J_2$ ) results for different network loads

RTT delay variability load described in Section 4.2. None of the six RTT delay distributions is centered at 35 ms, which is the delay distribution mean assumed in the design stage. This means that, for each distribution, the majority of the delays appearing in the networked control loop are unexpected. The analysis, which includes the four routes, evaluates the deviation and the traveling time.

Figures 5 and 6 provide  $J_1$  and  $J_2$  results, respectively, considering the different loads in the WiFi network. In order to simplify the figure, the results for the remote path tracking control approach using the mean value, max value, and medium value estimation techniques are integrated (averaged) into one evaluation, labeled *statistics*. As it is shown in the figures, as the load in the network increases, the remote path tracking control performance degradation increases. In the case of using the Kalman filter estimation algorithm, the remote path tracking control presents always the best performance compared with the other algorithms. Therefore, it presents a robust behavior in front of different network loads.

## 6 Conclusions

AGV path tracking control is an industrial application that can benefit from the advantages provided by wireless communication. However, closing the loop over a WiFi network has been shown to bring new challenges due to the negative effects that communication delays have in performance. This paper has presented a remote path tracking control algorithm where the use of a delay estimator becomes the key for successful operation. The proposed approach uses a simple stochastic model for the delay dynamics that permits to design an optimal estimator for the network delays. First, this solution has been shown to provide effective path tracking in terms of accuracy in following a trajectory and in terms of total traveling time. Second, the presented algorithm has been shown to be robust against unexpected WiFi delays distributions as well as against varying network load conditions. Third, results show that the remote path tracking control algorithm using the Kalman estimator outperforms previously presented WiFi delays estimation techniques

Future work will focus on minimizing the number of messages required for the remote AGV path tracking control system without compromising its operation. This will lead to control loops with less network requirements, as well as to a system with less energy demand.

## References

- Willig A, Matheus K, Wolisz A (2005) Wireless technology in industrial networks. *Proc IEEE* 93(6):1130–1151
- De Pellegrini F, Miorandi D, Vitturi S, Zanella A (2006) On the use of wireless networks at low level of factory automation systems. *IEEE Trans Ind Inf* 2(2):129–143
- Baturone I, Moreno-Velo FJ, Blanco V, Ferruz J (2008) Design of embedded DSP-based fuzzy controllers for autonomous mobile robots. *IEEE Trans Ind Electron* 55(2):928–936
- Tipsuwan Y, Chow MY (2004) Gain scheduling middleware: a methodology to enable existing controllers for networked control and teleoperation—part I: networked control. *IEEE Trans Ind Electron* 51(6):1218–1227
- Tipsuwan Y, Chow MY (2004) Gain scheduling middleware: a methodology to enable existing controllers for networked control and teleoperation—part II: teleoperation. *IEEE Trans Ind Electron* 51(6):1228–1237
- Tipsuwan Y, Chow MY (2003) Control methodologies in networked control systems. *Control Eng Pract* 11:1099–1111
- Yang TC (2006) Networked control system: a brief survey. *IEE Proc Control Theory Appl* 153(4):403–412
- Ye H, Walsh GC, Bushnell LG (2001) Real-time mixed-traffic wireless networks. *IEEE Trans Ind Electron* 48(5):883–890
- Kim WJ, Ji K, Ambike A (2006) Real-time Operating environment for networked control systems. *IEEE Trans Robot Autom* 3(3):287–296
- Martí P, Yépez J, Velasco M, Villá R, Fuertes JM (2004) Managing quality-of-control in network-based control systems by controller and message scheduling co-design. *IEEE Trans Ind Electron* 51(6):1159–1167
- Lian FL, Yook JK, Tilbury DM, Moyne J (2006) Network architecture and communication modules for guaranteeing acceptable control and communication performance for networked multi-agent systems. *IEEE Trans Ind Inf* 2(1):12–24
- Decotignie JD (2005) Ethernet-based real-time and industrial communications. *Proc IEEE* 93(6):1102–1117
- Ploplys NJ, Kawka PA, Alleyne AG (2004) Closed-loop control over wireless networks. *IEEE Control Syst Mag* 24(3):58–71
- Yoshizawa K, Hashimoto H, Wada M, Mori SM (1996) Path tracking control of mobile robots using a quadratic curve. In: *Proceedings of IEEE intelligent vehicle symposium*, pp 58–63
- Kalman RE (1960) A new approach to linear filtering and prediction problems. *Trans ASME J Basic Eng* 82:35–45
- Jacobsson K, Möller N, Johansson KH, Hjalmarsson H (2004) Some modeling and estimation issues in control of heterogeneous networks. In: *Proceedings of the 16th international symposium on mathematical theory of Networks and Systems*. Leuven, Belgium
- Lennvall T, Fohler G (2005) Providing adaptive QoS in wireless networks by traffic shaping. *Resource management for media processing in networked embedded systems*. Eindhoven, The Netherlands
- Li Z, Vanijirattikhan R, Chow MY, Viniotis Y (2005) Comparison of real-time network traffic estimator models in gain scheduler middleware by unmanned ground vehicle-based controller. In: *32nd annual conference of IEEE, industrial electronics society*
- Lozoya C, Martí P, Velasco M, Fuertes JM (2007) Effective real-time wireless control of an autonomous guided vehicle. In: *IEEE international symposium on industrial electronics*. Vigo, Spain

20. Specification of the Bluetooth System (2009) Version 1.1
21. IEEE LAN/MAN Standards Committee of the IEEE Computer Society (2003) IEEE Standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (LR-WPAN's). IEEE, New York
22. IEEE LAN/MAN Standards Committee of the IEEE Computer Society (1999a) Information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE, New York
23. IEEE LAN/MAN Standards Committee of the IEEE Computer Society (1999b) IEEE Standard for telecommunications and information exchange between systems—LAN/MAN specific requirements—part 11: wireless medium access control (MAC) and physical layer (PHY) specifications: high speed physical layer in the 5 GHz band. IEEE, New York
24. IEEE LAN/MAN Standards Committee of the IEEE Computer Society (1999c) IEEE Standard for information technology—telecommunications and information exchange between systems—local and metropolitan networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher speed physical layer (PHY) extension in the 2.4 Ghz band. IEEE, New York
25. Wiberg PA, Bilstrup U (2001) Wireless technology in industry—applications and user scenarios. In: Proceedings of the IEEE international conference on emerging technologies and factory automation, pp 123–131
26. Allman M, Paxson V (2001) On estimating end-to-end network path properties. *SIGCOMM Comput Commun Rev* 31(2):124–151
27. Gunawardena D, Key P, Massoulie L (2003) Network characteristics: modelling, measurements and admission control. In: Eleventh international workshop on quality of service. Monterey, CA
28. Gelb A (1974) *Applied optimal estimation*. MIT, Cambridge
29. Abu-Ali NA, Hassanein H (2006) WLC30-5: statistical delay budget partitioning in wireless mesh networks. In: IEEE global telecommunications conference