Adaptation of Rescue Robot Behaviour in Unknown Terrains Based on Stochastic and Fuzzy Logic Approaches

Ashraf Aboshosha
RA Dept.- Prof. Dr. A. Zell, WSI, University of Tübingen
Sand 1, D-72076 Tübingen, Germany
shosha@informatik.uni-tuebingen.de
http://www-ra.informatik.uni-tuebingen.de

***Abstract*— The purpose of this article is to provide rescue robots with an adaptive behaviour during searching for victims in disasters such as fire, earthquake, flood, wars etc. This experimental research work took place in previously unknown dynamic indoor terrains. The main phases of this framework are; 1) modelling of robot behaviours/dynamics in collapsed environments, 2) designing an adaptive controller, which regulates robot longitudinal velocity and heading (collision avoidance) based on the obstacles distribution histogram, 3) prediction of robot behaviours in another unknown terrain. Two approaches have been used to design the adaptive controller: the first one is the stochastic control theory, based on Kalman filter algorithms [10][8]. The second approach relies on fuzzy inference systems (FIS) [5][17][7]. Throughout this work, robot dynamics have been modelled using the auto regressive exogenous (ARX) scheme, while ARX model parameters have been identified using recursive least squares (RLS) [18]. This contribution presents a description and some discussion of the discrete Kalman filter, modelling techniques, and some discussion of robot behaviour analysis. Furthermore, the design of adaptive controllers using FIS-based techniques versus stochastic control systems has been demonstrated.** [1]

Keywords: *rescue robot, stochastic control, Kalman filter, fuzzy logic, adaptive navigation.*

I. Introduction

In disasters, autonomous mobile systems are highly needed to help in finding trapped victims. Intelligent mobile robots and cooperative robotic systems can be very efficient tools to speed up searching and rescue operations. Rescue robots usually have 48 hours to find trapped survivors. They are also useful to do rescuing jobs in situations that are hazardous for humans. Promising projects of rescue robots got recently a lot of attention worldwide, although the available technologies do not yet provide enough autonomy or precision as needed. Adaptive controllers compensate fluctuations of internal/external parameters of dynamic systems to preserve the stability and to increase the robustness. Based on the histogram of surrounding obstacles, the robot adapts its translational velocity and avoids collisions during exploration of the collapsed structure seeking for alive victims. Adaptive controllers reinforce rescue robot navigation safety by steering the robot behaviour smoothly to the desired set point, instead of abrupt variations that might cause a harmful effect on the internal structure of robots. In this study, stochastic Kalman filters have been employed to model robot dynamics, to identify model parameters, to predict robot behaviours in another terrain and to regulate the rescue robot velocity related to the free front space [16][14][4]. Kalman filters belong to algorithms (e.g. ARX, ARMA, LMS, RLS, FIR, IIR, LQR, LQG, etc.) that widely involved in adaptive systems to underlay subjects of modelling, identification, prediction, signal processing, behaviour learning and controller design [9][12]. On the other side, adaptive FIS techniques enable the robot to interact adaptively with the static or dynamic events during navigation. FIS have been successfully employed in automatic control, data classification, decision analysis, expert systems, time series prediction, and pattern recognition. Furthermore, FIS are considered in many robotic applications, such as velocity control, collision avoidance, path tracking, dynamic objects pursuing [1], map building and in sensor fusion [11]. In this paper, the following topics will be discussed: part (II) focuses on the formulation of discrete Kalman filter. Part (III) outlines signal and system modelling approaches, ARX model building, and using the RLS technique in identification of rescue robot model parameters. Part (IV) demonstrates how to regulate the robot velocity based on the obstacles distribution histogram using adaptive Kalman filters and practical results are presented. Part (V) illustrates applying FIS techniques to regulate robot behaviours. Part (VI) introduces a comparison between stochastic based and FIS based control approaches.

[1]This study has been implemented on the B21-RWI robot platform (Colin and Robin - laboratory for autonomous mobile robots, University of Tübingen).

II. Discrete Kalman Filter

Kalman filters are widely used in studies of dynamic systems, analysis, estimation, prediction, processing and control. The Kalman filter is an optimal solution for the discrete data linear filtering problem. The Kalman filter is a set of mathematical equations that provides an efficient computational solution to sequential systems. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states (prediction), and it can do so even when the precise nature of the modelled system is unknown. The filter is derived by finding the estimator for a linear system, subject to additive white Gaussian noise [10]. The discrete form of Kalman filters addresses the general problem of trying to estimate the state of a discrete time process that is governed by the linear stochastic difference equation. The Kalman filter formulation begins with definition of the system and assumptions. Consider a process f(.) given by the state space model defined by its linear difference equation form

$$\upsilon_{k+1} = f(\upsilon_k, \lambda_k, \xi_k) = A_k \upsilon_k + B_k \lambda_k + \xi_k \tag{1}$$

The observation or measurement model is a mapping of the actual system state υ to an observed state y. The system is driven by a control signal λ, while measurements are associated with Gaussian white noises ξ and γ. The discrete time observation model is given by the linear difference equation

$$y_k = f(\upsilon_k, \gamma_k) = S_k \upsilon_k + \gamma_k \tag{2}$$

In our case the observed output is the rescue robot longitudinal velocity/heading and the reference control signal is the obstacles distribution histogram. The coefficients A, B and S are Kalman filter parameters.

III. Robot Dynamics Modelling

A. Signal and System Modelling

There are two different approaches to the characterization of dynamic systems: In linear systems theory, one can assume either some structure in the signals or some structure in the system. Attempts have been made to combine these two approaches e.g. harmonic identification techniques in the Fourier domain.

First approach: Structure the signal can be found using linear transforms. This approach does not take into account that the system has some structure. In the time domain, filtering is a linear transformation. The Fourier, Wavelet, and Karhunen-Loeve transforms have compression Capability and can be used to identify some structure in the signals. When we are using these transforms, we do not take into account any structure in the system.

Second approach: Structure the system can be found by fitting a model to the system.

B. Approaches to System Modelling

Physical models of robots are either reduced-size copies of the original dynamics following the laws of model similarity, or analogies. The idea of an analogy implies that there exists ”something” at every instant of time that is to be analogous to the dependent variables of the original physical system. Mathematical models map the relationships between the physical variables in the robot dynamics to be modelled onto mathematical structures like simple algebraic equations, systems of differential equations or even difference equations. Mathematical modelling of robots can be developed in different ways: either purely theoretically based on the physical relationships (sensors actuators Interaction), which are a priori known about the robot dynamics, or purely empirically by experiments on the already existing robot, or by a sensible combination of both ways. Models obtained by the first method are often called priori, first principle or theoretical models, while models obtained in the second way are called posteriori or experimental models. Theoretical model building becomes unavoidable if experiments in the respective system cannot or must not be carried out. If the system to be modelled does not yet exist, theoretical modelling is the only possibility to obtain a mathematical model.

C. ARX Modelling

The discrete ARX modelling scheme is derived from Kalman filter, see figure (1). The ARX scheme is widely used in modelling of sequential system dynamics. This structure takes into account both the observed state υ_k and the driving control signal λ_k which is given by:

$$\upsilon_k = \sum_{i=1}^{n_a} a_i \upsilon_{(k-i)} + \sum_{i=0}^{n_b} b_i \lambda_{(k-i)} + \eta_k \tag{3}$$

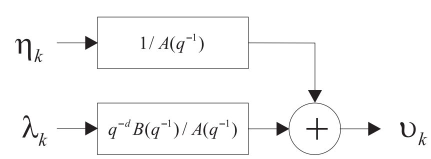

Fig. 1. ARX modelling of rescue robot dynamics

Where; η is the modelling residual, representing the white noise. n_a is the model order of the observed state (also called the number of poles). n_b is the model order of the control signal (also called the number of zeros). The operator q^{-1} is the back shift operator or delay, which is given by $q^{-1}\upsilon_k = \upsilon_{(k-1)}$, that follows:

$$\begin{aligned} A(q^{-1})\upsilon_k &= q^{-1}B(q^{-1})\lambda_k + \eta_k \\ A(q^{-1}) &= 1 - a_1 q^{-1} - a_2 q^{-2} - \ldots - a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{n_a} q^{-n_b} \end{aligned} \tag{4}$$

The observed state υ_k is the longitudinal velocity/heading while the control signal λ_k is the obstacles distribution histogram, acquired by laser/sonar. The Gaussian distributed noise, associated with the observed output, allows applying identification algorithms such as; RLS or least mean squares (LMS) to estimate model parameters (coefficients). This modelling scheme is only applicable within linear or quasi-linear systems. Therefore, it is applied within this framework to regulate the velocity/heading, while this algorithm failed to cope with position control of mobile robots due to enormous non-linear odometric errors [2].

D. RLS Estimation of Model Parameters

Now let us explain, how to estimate ARX model parameters $A(q^{-1})$ and $B(q^{-1})$. The RLS is a stepwise learning algorithm, this means that, the estimation of model parameters has a gradual convergence. Compared with the LMS algorithm (batch-wise learning), the RLS needs a lower computational power and it is more stable than the LMS. The LMS formulation depends on the matrix inversion and some matrices are not invertible. We can initialize the RLS identification process using an empty vector. Also, we can initialize it using fuzzy and neural nets. Figure (2) shows outputs of both the ARX model and an actual robot output. The ARX model output is smooth due to filtering of high frequencies. Figure (7) shows the convergence of parameters during the learning process.

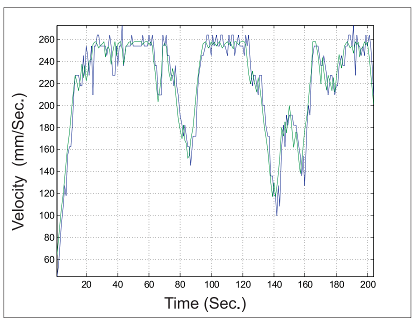

Fig. 2. Output of both the rescue robot and its ARX model

The RLS estimation process can be accomplished according to the following. At time step $(k+1)$:

1) Form $\Gamma_{(k+1)}$using the new data (input-output patterns)

$$\Gamma^T_{(k+1)} = [\upsilon_{(k+1)}....\upsilon_{(k-n_a)}\lambda_{(k)}.........\lambda_{(k-n_b)}] \quad (5)$$

2) Form the estimation error $\varepsilon_{(k+1)}$ using

$$\varepsilon_{(k+1)} = \upsilon_{(k+1)} - \overline{\upsilon}_{(k+1)} = \upsilon_{(k+1)} - \Gamma^T_{(k+1)}\Theta_{(k)} \quad (6)$$

3) Form $\Psi_{(k+1)}$ (based on Kalman gain) using

$$\Psi_{(k+1)} = \Psi_{(k)}\left[I_m - \frac{\Gamma_{(k+1)}\Gamma^T_{(k+1)}\Psi_{(k)}}{(1+\Gamma^T_{(k+1)}\Psi_{(k)}\Gamma_{(k+1)})}\right] \quad (7)$$

4) Update parameters

$$\Theta_{(k+1)} = \Theta_{(k)} + \Psi_{(k+1)}\Gamma_{(k+1)}\varepsilon_{(k+1)} \quad (8)$$

where

$$\Theta^T_{(k+1)} = [1 - a_1..... - a_{n_a} + b_0 + ... + b_{n_b}] \quad (9)$$

5) Wait for the next time step to elapse and loop back to step (1)

IV. Velocity Control Based on Kalman Filter

Applying adaptive control theories to autonomous robots leads to a smooth transition among operation levels. Moreover, these systems own a deliberative structure, which involves reference signal generation, modelling, identification, controller design and sometimes optimizing units, see figure (3). To generate a reference signal, environment maps have been built using two B21-RWI robots, equipped with 24 Polaroid 6500 sonar sensor and a Sick LMS 200 laser range finder. The characteristics, advantages and disadvantages of map building using both of time of flight (TOF) sensors (laser and sonar) can be reviewed in [13] and [2]. To build an obstacles histogram of an environment, different methods can be used. There are: 1) nearest obstacle of the front free space, 2) area of the surrounding free space or 3) the free front space area. What has been used throughout this work, is the nearest obstacle of the free front space. Figures (4, 5) present two models of map building using the laser range finder and sonar modules [2]. The implementation of the adaptive

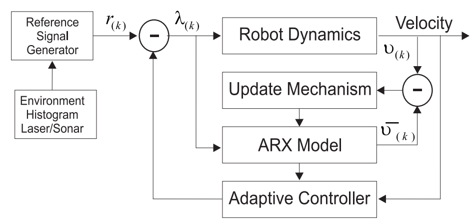

Fig. 3. Adaptive control of rescue robot longitudinal velocity

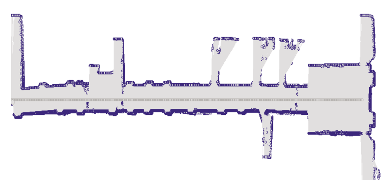

Fig. 4. Map building using the Sick LMS 200 laser scanner

Fig. 5. Map building using Polaroid 6500 sonar sensors

control starts with data acquisition of input/output patterns (histogram/velocity). Then, ARX paradigm depicts rescue robot dynamics. Meanwhile, the RLS algorithm identifies ARX Model parameters (coefficients). After training, the ARX model can resemble the rescue robot behaviour with another histogram. Figure (2) shows ARX model learning using the RLS algorithm. The model output is highly correlated to the reference signal (after omitting high frequency noises). Figure(6) presents the prediction of rescue robot behaviour driven by arbitrary histogram. The convergence of model parameters during learning process is presented in figure(7), this convergence reflects the success of training. Figure (8) show white noise associated with learning process. In adaptive systems, the design of

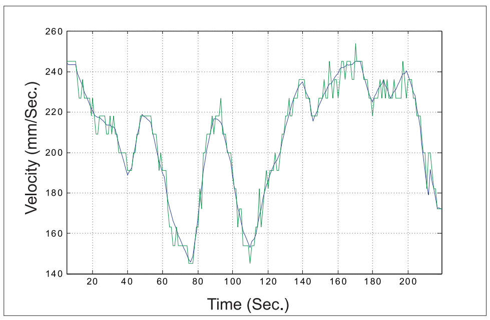

Fig. 6. Output of ARX model in prediction mode

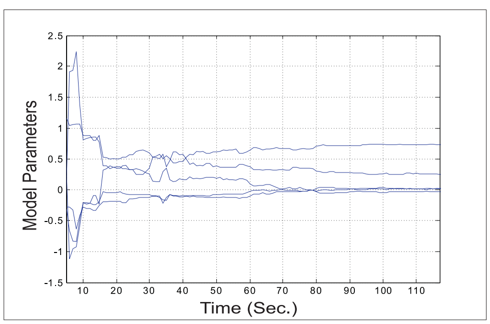

Fig. 7. Convergence of ARX model Parameters

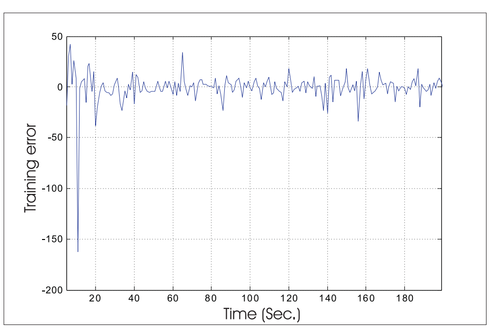

Fig. 8. Training error

controllers relies on on-line estimated model parameters. Throughout this phase, the pole assignment self-tuning controller forces the robot behaviour to yield a definite pattern [4][3]. The controller compensates not only the transient time errors but also steady state scalar errors according to the following:

$$\frac{\upsilon_k}{r_k} = q^{-d}.\frac{B(q^{-1})C}{T(q^{-1})} \text{ and } C = \lim_{q\to 1}\frac{T(q^{-1})}{B(q^{-1})} \qquad (10)$$

where, $T = 1 + t_1 z^{-1} + ... + t_{n_t} z^{-n_t}$ is the assigned pole function and C is the compensator. To choose appropriate controller parameters, coefficients of the robot dynamics model have to be incorporated in computational processes. Successful controller parameters achieve the maximum correlation (ρ) between the observed state υ_k and the reference signal r_k. The delay factor d is considered as unity for simplicity.

$$\rho = \frac{\text{cov}(\upsilon, r)}{\sqrt{Var(\upsilon)}\sqrt{Var(r)}} \qquad (11)$$

Figure (9) shows the variation of the correlation factor ρ due to changing desired (assigned) poles T.

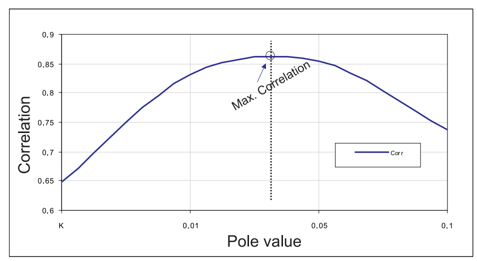

Fig. 9. The relation between correlation and controller parameters

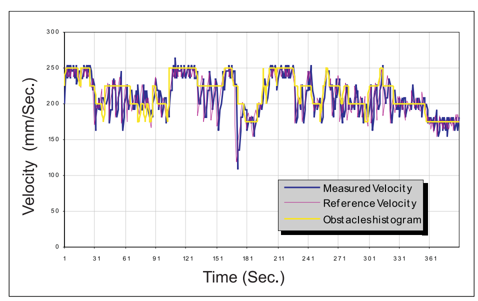

Fig. 10. Adaptive pole assignment control of robot velocity

V. Fuzzy Controller

In 1965, Zadeh published his paper (Fuzzy Sets). After that scientists worldwide developed different algorithms to design a fuzzy logic controller (FLC) e.g.; E. Mamdani 1975, Takagi-Sugeno 1985 and Tsukamoto fuzzy model. FIS present a considerable solution to the subject of mobile robots control [6]. The fundamental three phases of FLC are; fuzzification, inference engine design and defuzzification. These three phases are analogous to three phases of stochastic based control systems; modelling, identification and controller design. A FLC is an intelligent control system that smoothly interpolates between rules. A fuzzy set may be represented by a mathematical formulation known as a membership function. That is, associated with a given linguistic variable (e.g. mobile robot velocity) are linguistic values or fuzzy subsets (e.g. slow, fast, etc.), expressed as membership functions, which represent uncertainty, vagueness, or imprecision in values of the linguistic variable, see figure (11). This function

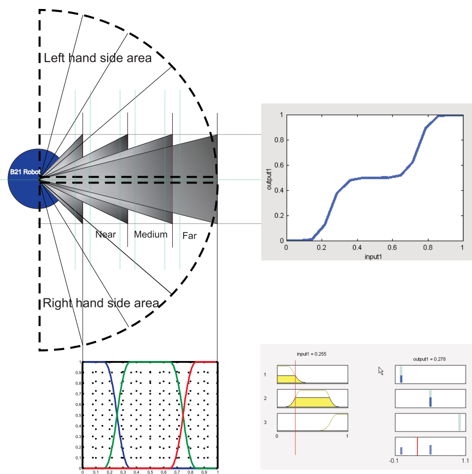

Fig. 11. Fuzzy control of robot's longitudinal velocity

assigns a numerical degree of membership, in the closed unit interval, to a crisp (precise) number. Within this framework, a membership value of zero-one corresponds to an element that is definitely not/definitely a member of the fuzzy set. Partial membership is indicated by values between 0 and 1. Implementation of a fuzzy controller requires assigning membership functions for inputs and outputs. Inputs are usually measured variables, associated with the state of the controlled plant that are assigned membership values before being processed by an inference engine. The heart of the controller inference engine is a set of if-then rules whose antecedents and consequents are made up of linguistic variables and associated fuzzy membership functions. Fuzzy set intersection, or conjunction, operators in the antecedent are generally referred to as t-norms. They commonly employ algebraic min or product operations on fuzzy membership values. Consequents from different rules are numerically aggregated by fuzzy set union and then defuzzified to yield a single crisp output as the control for the plant [15]. The most popular FLC algorithm is the discrete Takagi-Sugeno fuzzy model, the consequent part of the rules is described by nonfuzzy analytical functions. The discrete FIS considered in this paper is defined by the following implications:

1) The centroid or the centre of gravity (COG) defuzzification rule g is expressed by:

$$g = \frac{\sum_{i=1}^{n} \omega_i g_i}{\sum_{i=1}^{n} \omega_i} \tag{12}$$

2) Formulating the state representation as follows

$$g_{(k+1)} = \frac{\sum_{i=1}^{n} \omega_{(i,k)} (A(q^{-1}) g_{(k)} + B(q^{-1}) \upsilon_{(k)})}{\sum_{i=1}^{n} \omega_{(i,k)}} \tag{13}$$

3) Driving the controller observed state $o_{(k)}$

$$o_{(k)} = \frac{\sum_{i=1}^{n} \omega_{(i,k)} S(q^{-1}) g_{(k)}}{\sum_{i=1}^{n} \omega_{(i,k)}} \tag{14}$$

4) Calculating the robot control signal $\lambda_{(k)}$

$$\begin{aligned}\lambda_{(k)} &= r_{(k)} - F(q^{-1}) o_{(k)} \\ &= r_{(k)} - \frac{\sum_{i=1}^{n} \omega_{(i,k)} F(q^{-1}) S(q^{-1}) o_{(k)}}{\sum_{i=1}^{n} \omega_{(i,k)}}\end{aligned} \tag{15}$$

Figure 12 shows how the robot's output follows the reference signal smoothly, highly correlated (the cross correlation of both υ and r is 0.936). Compared with results of pole assignment control, shown in figure (10) (the cross correlation of both υ and r is 0.7367), the FLC has a better performance. Based on the previous

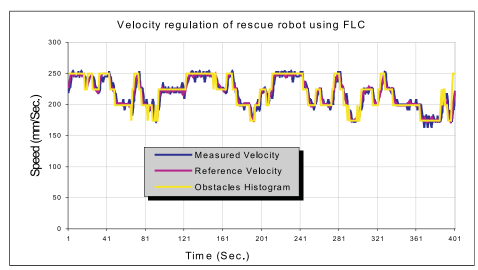

Fig. 12. Adaptive regulation of rescue robot velocity using FLC

results, we can find that the FLC can deal with model based and behaviour based control systems. In collision avoidance, FIS have a better performance than what has been achieved by Kalman filters. The random dynamic obstacles have no definite motion model, therefore Kalman filters failed to cope with them, while FIS succeeded to interact adaptively with those events. On the other side, the model of robot velocity has a clear modelling scheme, which can be easily handled using Kalman filters. We have to take into consideration that the COG function fails to deal with some types of membership functions. The major differences between both approaches are explained in details in the next part.

VI. Kalman Filter versus Fuzzy Logic

The conclusion of this contribution is introduced as a comparison between stochastic based and FIS based control systems. The following points distinguish between both of them:

- Stochastic control systems are applicable only to well-structured problems such as feedback control of linear or piecewise linear dynamic systems where our knowledge about the problems is deep and extensive. FIS do not need models to control a system.
- Almost all applications of FIS to traditional system problems amount to nothing more than interpolating or extrapolating among well-known controller designs implemented with stochastic control techniques. However, the world is full of control problems beyond this narrow confine.
- Most problems are poorly understood and described only in natural language terms. For these problems, FIS can play a role either by quantifying imprecise natural language and/or by converting human experience to systematic but logic if-then rules.
- The difference between logic and stochastic system is simply an illustration of generality versus depth.
- Stochastic control system analysis tends to be sophisticated compared with the simplicity of FIS.
- Contrary to the stochastic control system, FIS pay a poor reaction to the fully interconnection and to the parallelism.
- Logic computing systems support the independent pattern recognition, while stochastic systems support behavior learning.

Acknowledgements

I would like to acknowledge the financial support by the German Academic Exchange Service (DAAD) of my PhD scholarship at the University of Tübingen.

VII. References

[1] Ashraf Aboshosha. Adaptive visual guidance of mobile robots based on takagi-sugeno model. In *proceedings of Applied Machine Vision - from Image Processing to Vision Systems*, Stuttgart, Germany, 21.-22. October 2003. VDI.

[2] Ashraf Aboshosha and Andreas Zell. Robust mapping and path planning for indoor robots based on sensor integration of sonar and a 2d laser range finder. In *INES 2003*, Assiut-Luxor, Egypt, March 2003.

[3] K. J. Astrom and Bjorn Wittenmark. *Computer-Controlled Systems, Theory and Design*. Prentice Hall, New Jersey, USA, 3 edition, 1997.

[4] F. Calugi, A. Robertsson, and R. Johansson. Output feedback adaptive control of robot manipulators using observer backstepping. In *Proceeding of IROS*, EPFL Lausanne, Switzerland, 2002. IEEE.

[5] D. Driankov and A. Saffiotti. *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Physica-Verlag, 2001.

[6] J. T. Economon, A Tsourdos, and B. A. White. Takagi-sugeno model synthesis of a quasi-linear parameter varying mobile robot. In *IROS*, EPFL, Lausanne, Switzerland, 2002. IEEE.

[7] S. G. Goodridge et.al. Multi-layered fuzzy behaviour fusion for reactive control of an autonomous mobile robot. In *Proceeding of the 6th International Conf on Fuzzy Systems*, Barcelona, SP, 1997. IEEE.

[8] R. Isermann, K.-H. Lachmann, and D. Matko. *Adaptive Control Systems*. 1992.

[9] M. Kalandros and L. Y. Pao. Covariance control for multisensor systems. In *IEEE Trans. Aerospace Electronic Systems*, October 2002.

[10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME Journal of Basic Engineering*, pages 35–45, MARCH 1960.

[11] F. Kobayashi, Y. Tanabe, T. Fukuda, and F. Kojima. Acquisition of sensor fusion rule based on environmental condition in sensor fusion system. In *Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS Int'l Conf.*, pages 2096–2101, 2001.

[12] M.Juan, Martn Snchez, and Jos Rodellar. Adaptive predictive control: From the concepts to plant optimization. 1996.

[13] A. Mojaev and A. Zell. Sonardaten-integration fuer autonome mobile roboter. In *Tagungsband zum 20. DAGM-Symposium Mustererkennung*, pages 556–563, Stuttgart - Germany, October 1998.

[14] T.V.Guy, M. Krn, and J. Bhm. Linear adaptive controller based on smoothing noisy data algorithm. In *Proceedings of European Control Conference ECC*, Karlsruhe, Germany, 1999.

[15] G. Vachkov and T. Fukuda. Structured learning and decomposition of fuzzy models for robotic control applications. *J. of Intelligent and Robotic Systems*, 2001.

[16] A. Varga. Robust pole assignment techniques via state feedback. In *Proceedings of IEEE Conference on Decision and Control, CDC*, Sydney, Australia, 2000.

[17] O. Popovici Vlad and T. Fukuda. Design method for neuro-fuzzy motion controllers. In *Proc. of the International Conference on Industrial Technology ICIT*. IEEE, 2002.

[18] P.E. Wellstead and M.B. Zarrop. *Self-tuning Systems Control and Signal Processing*. Wiley, 1991.