Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Optimal Membership Functions For Multi-Modal Control

Tejas R. Mehta and Magnus Egerstedt

Abstract— **In order to manage the rapidly growing complexity associated with many modern control applications, multi-modal control has emerged as a viable option. In this paper, we assume that an ensemble of individual controllers or modes have been designed, and concentrate on the problem of combining these controllers in an optimal manner. In particular, we use variational arguments for shaping parameterized membership functions in order to minimize a given performance cost, thus resulting in a systematic method for optimal mode fusion in a general setting. This view complements the standard fuzzy logic control approach, where the modes are combined according to some given (and possibly fine-tuned) membership functions. Moreover, the new controller fusion methodology is transitioned onto a real robotic platform navigating in an unknown environment to compare the performance of the proposed method with the standard approach.**

I. INTRODUCTION

Multi-modal control is a commonly used design tool for breaking up complex control tasks into sequences of simpler tasks. The main idea is to design a collection of modes, or control laws, defined with respect to a particular task, data source, or operating point, and then concatenate these control laws in order to produce the desired overall behavior. Given that such a mode string is the design objective, the control task thus involves mapping symbols (tokenized mode descriptions) to signals rather than signals (control values) to signals. A number of modelling paradigms facilitating this construction have been proposed from *Hybrid Automata* [1] to *Motion Description Languages* [2], [3].

In this paper, we assume that the individual controllers or modes have already been designed, and concentrate on the problem of combining these controllers in an optimal manner. Note we are not trying to obtain the optimal modal sequence, but rather we will construct a new controller that is an optimal aggregation of the existing modes. This is an area that has received considerable attention. The novelty of this paper comes from the use of variational methods for mode fusion in the general setting. This view complements the standard approach, where the modes are combined according to some given (and possibly fine-tuned) membership functions, as is the case in fuzzy logic control. For a representative sample, see [4], [5], [6].

The outline of this paper is as follows: In Section II the background to the problem is given, followed by its variational solution in Section III. Section IV introduces the notion of membership functions in fuzzy control, followed by a robotics example in Section V and conclusions in Section VI.

This work was supported by DARPA through Grant number FA8650-04-C-7131

T. R. Mehta is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA tmehta@ece.gatech.edu

M. Egerstedt is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA magnus@ece.gatech.edu

II. PROBLEM FORMULATION

Formally, we define a mode σ as a pair (κ, ξ), where $\kappa : X \to U$ corresponds to a particular feedback law, and the interrupt $\xi : X \to \{0, 1\}$ encodes conditions for its termination. Note here that X, U denote the state and input space respectively. Given a finite set of feedback mappings K and interrupts Ξ, we let $\Sigma = K \times \Xi$ denote the set of all modes, or control-interrupt pairs. Moreover, by $\Sigma^\star$ we understand the set of all finite length strings over Σ, while Σ^N is the set of strings with length less than or equal to N. Now let $X_R^{\Sigma^\star} \subset X$ denote the reachable subset of X induced by $\Sigma^\star$. The calculation of the reachable subset has been thoroughly studied and there is an abundance of literature pertaining to the estimation of $X_R^{\Sigma^\star}$. To name a few, [7], [8], [9] proposed analytical methods for achieving this, while [10], [11], [12] concerned the development of numerical algorithms for computing the reachable set. In particular, in [13] Rapidly-exploring Randomized Trees (RRTs) [14], [15] were employed to estimate the reachable set. Now, if we bound the length of the control program (i.e. $|\bar{\sigma}| \leq N$, where $\bar{\sigma} = \sigma_1\sigma_2 \cdots \sigma_k$ ($k \leq N$) is the control sequence that transfers the system from x_0 to a state $x \in X_R^{\Sigma^N}$) and fix x_0, then the set $X_R^{\Sigma^N}$ is finite and we can find the optimal mode string $\bar{\sigma}^* \in \Sigma^N$ that minimizes some given cost $J(x, \bar{\sigma})$, e.g. using reinforcement learning [16], [17].

Once we have obtained $\bar{\sigma}^*$, what new modes, if any, should be added in order to improve the performance? In order to answer this question, we propose to add new modes in a highly structured manner by merging or combining recurring mode sequences into one smooth mode, if possible. Suppose for example there are multiple occurrences of $\sigma_1 = (\kappa_1, \xi_1)$ followed by $\sigma_2 = (\kappa_2, \xi_2)$ in our control program $\bar{\sigma}$ (i.e. $\bar{\sigma} = \sigma_1\sigma_2\sigma_3\sigma_1\sigma_2\sigma_4\sigma_1\sigma_2$), we would like to replace this mode pair $(\sigma_1\sigma_2)$ with one single mode $\sigma_{new} = (\kappa_{new}, \xi_2)$ that produces the combined behavior of $\sigma_1\sigma_2$, to some degree of accuracy. Hence, defining this new mode is equivalent to defining a new feedback law κ_{new}. In order to manage the complexity, we constrain κ_{new} to be a function of the existing feedback laws, i.e. $\kappa_{new} = \delta(\kappa_1, \kappa_2, \ldots, \kappa_{card(K)})$ for some $\delta : K^{card(K)} \to (X \to U)$.

In [18], the new feedback law was constrained to be a linear combination of the existing functions. In particular, the weights for each feedback law in the linear combination were constant. In contrast, in this paper we let these weights be determined by a general function, $\mu(x,\alpha) : X \times \mathbb{R}^m \to \mathbb{R}$, parameterized by α. We will refer to these weighting functions as *membership functions*, as they loosely resemble membership functions used in fuzzy control. This problem can in fact be posed as a general optimization problem and solved using the calculus of variations.

In particular, assume that the system dynamics is given by

$$\dot{x} = f(x,u), \tag{1}$$

and that mode i is defined such that

$$\dot{x} = f(x, \kappa_i(x)) \equiv f_i(x) \tag{2}$$

until time $t = \tau_i$, i.e. $\xi_i(x(\tau_i)) = 1$. Now given the switched control system

$$\dot{x}(t) = \begin{cases} f_1(x(t)) & \text{if } t \in [0, \tau_1] \\ f_2(x(t)) & \text{if } t \in [\tau_1, \tau_2] \\ \vdots & \vdots \\ f_M(x(t)) & \text{if } t \in [\tau_{M-1}, T] \end{cases}, \tag{3}$$

where $x \in \mathbb{R}^n$, $x(0) = x_0$, and T is given final time. Here $f_i : \mathbb{R}^n \to \mathbb{R}^n$ for $i = 1, \ldots, M$, are continuously differentiable functions. Now assume we have a set of continuously differentiable functions $g_i : \mathbb{R}^n \to \mathbb{R}^n$ and $\mu_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for $i = 1, 2, \ldots, M$, and let

$$\dot{z}(t) = \sum_{i=1}^{N} \mu_i(z(t), \alpha_i) g_i(z(t)), \tag{4}$$

where $z \in \mathbb{R}^n$ and $z(0) = x_0$. Here $g_i = \zeta(f_1, f_2, \ldots, f_M)$ and μ_i is a membership function on $\mathbb{R}^n$ that is parameterized by vector α_i, e.g. $\mu(z,\alpha) = ae^{-b\|z-z_0\|^2}$ where $\alpha = [a,b]^T$. Note that μ must be continuously differentiable with respect to both z and α. We want to choose $\alpha_1, \alpha_2, \ldots, \alpha_N$ so that the cost function

$$J(\alpha_1, \ldots, \alpha_N) = \int_0^T L(x(t), z(t))dt + \psi(x(T), z(T)) \tag{5}$$

is minimized, where L and ψ are continuously differentiable in their second argument.

III. VARIATIONAL SOLUTION

By adding the constraint with a co-state $\lambda(t)$ to (5), we obtain

$$\begin{aligned} \tilde{J} \equiv \tilde{J}(\alpha_1, \ldots, \alpha_N) = \int_0^T \Big[& L(x(t), z(t)) + \\ & + \lambda(t) \Big(\sum_{i=1}^{N} \mu_i(z(t), \alpha_i) g_i(z(t)) - \dot{z}(t) \Big) \Big] dt + \\ & + \psi(x(T), z(T)). \end{aligned} \tag{6}$$

Note above that $\tilde{J}$ denotes the unperturbed cost. Now we perturb (6) in such a way that $\alpha_k \to \alpha_k + \epsilon \vec{\theta}_{k_l}$, where $\vec{\theta}_{k_l} = [0, \ldots, \theta_{k_l}, \ldots, 0]^T$ (note the l^{th} entry is θ_{k_l} and all other entries are 0's), and $\epsilon << 1$, then $z \to z + \epsilon\eta$ is the resulting variation in $z(t)$. Note that above, we dropped the argument t when referring to $z(t)$ and will continue this convention in the following development for compactness with the implicit understanding that x, z and λ are functions of t. Now the perturbed cost is given by

$$\begin{aligned} & \tilde{J}_\epsilon \equiv \tilde{J}(\alpha_1, \ldots, \alpha_k + \epsilon\vec{\theta}_{k_l}, \ldots, \alpha_N) = \\ & = \int_0^T \Big[L(x, z+\epsilon\eta)) + \lambda \Big(\mu_1(z+\epsilon\eta, \alpha_1) g_1(z+\epsilon\eta) + \\ & \quad + \ldots + \mu_k(z+\epsilon\eta, \alpha_k + \epsilon\vec{\theta}_{k_l}) g_k(z+\epsilon\eta) + \ldots + \\ & \quad + \mu_N(z+\epsilon\eta, \alpha_N) g_N(z+\epsilon\eta) - \dot{z}(t) - \epsilon\dot{\eta} \Big) \Big] dt + \\ & \quad + \psi(x(T), (z+\epsilon\eta)(T)). \end{aligned} \tag{7}$$

Hence the Gateaux (also referred to as directional) derivative of $\tilde{J}$ in the direction of $\vec{\theta}_{k_l}$ is

$$\begin{aligned} \nabla_{\vec{\theta}_{k_l}} \tilde{J}(\alpha_1, \ldots, \alpha_N) &= \lim_{\epsilon \to 0} \frac{\tilde{J}_\epsilon - \tilde{J}}{\epsilon} \\ &= \int_0^T \Big[\frac{\partial L}{\partial z}\eta + \sum_{i=1}^{N} \lambda (\mu_i \frac{\partial g_i}{\partial z} + g_i \frac{\partial \mu_i}{\partial z})\eta + \\ & \quad + \lambda \frac{\partial \mu_k}{\partial \alpha_{k_l}} g_k(z)\theta_{k_l} - \lambda\dot{\eta} \Big] dt + \frac{\partial \psi}{\partial z}\eta(T), \end{aligned} \tag{8}$$

where α_{k_l} denotes the l^{th} element of vector α_k. By integrating $\lambda\dot{\eta}$ in (8) by parts and rearranging terms, we obtain

$$\begin{aligned} \nabla_{\vec{\theta}_{k_l}} \tilde{J} = \int_0^T & \Big[\frac{\partial L}{\partial z} + \lambda \sum_{i=1}^{N} (\mu_i \frac{\partial g_i}{\partial z} + g_i \frac{\partial \mu_i}{\partial z}) + \dot{\lambda} \Big] \eta dt + \\ & + \theta_{k_l} \int_0^T \frac{\partial \mu_k}{\partial \alpha_{k_l}} \lambda g_k(z) dt - \Big[\lambda\eta \Big]_0^T + \frac{\partial \psi}{\partial z}\eta(T) \end{aligned} \tag{9}$$

Note that $\eta(0) = 0$ since $z(0) = x(0) = x_0$. Now choose

$$\lambda(T) = \frac{\partial \psi}{\partial z} \tag{10}$$

$$\begin{aligned} \dot{\lambda}(t) = -\frac{\partial L}{\partial z}(x,z) - \lambda(t) \sum_{i=1}^{N} \big(& \mu_i(z, \alpha_i) \frac{\partial g_i}{\partial z}(z) + \\ & + g_i(z) \frac{\partial \mu_i}{\partial z}(z, \alpha_i) \big) \end{aligned} \tag{11}$$

With this choice of co-state $\lambda(t)$, which can be solved by integrating (15) backwards with initial condition (10), we obtain

$$\nabla_{\vec{\theta}_{k_l}} \tilde{J} = \Big[\int_0^T \frac{\partial \mu_k}{\partial \alpha_{k_l}}(z, \alpha_i) \lambda(t) g_k(z(t)) dt \Big] \theta_{k_l} \tag{12}$$

Finally, note that (12) gives access to the partial derivative $\frac{\partial \tilde{J}}{\partial \alpha_{k_l}}$ since we know that

$$\nabla_{\vec{\theta}} \tilde{J} = \frac{\partial \tilde{J}}{\partial \alpha_{1_1}} \theta_{1_1} + \frac{\partial \tilde{J}}{\partial \alpha_{1_2}} \theta_{1_2} + \ldots + \frac{\partial \tilde{J}}{\partial \alpha_{N_l}} \theta_{N_l}, \tag{13}$$

where $\vec{\theta} = [\theta_{1_1}, \theta_{1_2}, \ldots, \theta_{N_l}]^T$. Hence using (12) , (13), and the fact that α_k's are independent of each other, we deduce that

$$\frac{dJ}{d\alpha_{k_l}} = \int_0^T \frac{\partial \mu_k}{\partial \alpha_{k_l}}(z, \alpha_i) \lambda(t) g_k(z(t)) dt. \tag{14}$$

We summarize these results in the following theorem:

Theorem *Given a function $x(t) \in \mathbb{R}^n$ and a set of continuously differentiable functions $g_i : \mathbb{R}^n \to \mathbb{R}^n$ and $\mu_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for $i = 1, 2, \ldots, N$, with $z(t) \in \mathbb{R}^n$ given by (4), an extremum to the cost function*

$$J(\alpha_1, \ldots, \alpha_N) = \int_0^T L(x(t), z(t))dt + \psi(x(T), z(T))$$

is attained when the control vector $\vec{\alpha} = [\alpha_{1_1}, \alpha_{1_2}, \ldots, \alpha_{N_l}]^T$ is chosen such that

$$\frac{dJ}{d\alpha_{k_l}} = \int_0^T \frac{\partial \mu_k}{\partial \alpha_{k_l}} \lambda(t) g_k(z(t))dt = 0 \ \text{for } k = 1, \ldots, N,$$

where the co-state $\lambda(t)$ is chosen as follows:

$$\lambda(T) = \frac{\partial \psi}{\partial z}$$

$$\dot{\lambda}(t) = -\frac{\partial L}{\partial z}(x, z) - \lambda(t) \sum_{i=1}^{N} \big(\mu_i(z, \alpha_i) \frac{\partial g_i}{\partial z}(z) + + g_i(z) \frac{\partial \mu_i}{\partial z}(z, \alpha_i)\big).$$

The motivation for obtaining an expression for the gradient (14) is that we can employ a gradient descent method. The following numerical algorithm is proposed:

At each iteration i, where $\vec{\alpha}^{(i)}$ is the current vector of control variables, we follow these steps :

1) Compute the approximation function $z(t)$ forward in time from 0 to T using (4).
2) Compute the co-state $\lambda(t)$ backward in time from T to 0 using (10) and (15).
3) Compute the gradient $\nabla \tilde{J}(\vec{\alpha}^{(i)}) \Big[\frac{\partial \tilde{J}}{\partial \alpha_{1_1}^{(i)}}, \ldots, \frac{\partial \tilde{J}}{\partial \alpha_{N_l}^{(i)}} \Big]^T$ using (14).
4) Update the control variables as follow :

$$\vec{\alpha}^{(i+1)} = \vec{\alpha}^{(i)} - \gamma^{(i)} \nabla \tilde{J}(\vec{\alpha}^{(i)})$$

Note that the choice of the step-size $\gamma^{(i)}$ can be critical for the method to converge. An efficient method among others is the use of Armijo's algorithm, presented in [19]. Because of the non-convex nature of the cost function J, this gradient descent algorithm will only converge to a local minimum. Hence the attainment of a "good" local minimum can be quite dependent on the choice of a "good" initial guess for the control variables. However, the method presented here can still offer significant reductions in the cost function, as we will show experimentally. The association of such a local method with heuristic strategies in order to find a global minimum is not investigated here.

IV. MEMBERSHIP FUNCTIONS IN FUZZY-LOGIC CONTROL

Fuzzy-logic control is a method of defining nonlinear rule-based control systems. Although there are many different types of fuzzy controllers, we will focus on the popular Takagi and Sugeno (TS) fuzzy controller [4]. In the TS approach, the inputs are specified as fuzzy sets (fuzzification) as done in the standard approach, but the decision logic provides a crisp output so there is no defuzzification process. The architecture of a TS fuzzy controller is shown in Figure 1. There is typically an ensemble of control laws, i.e. $u =$

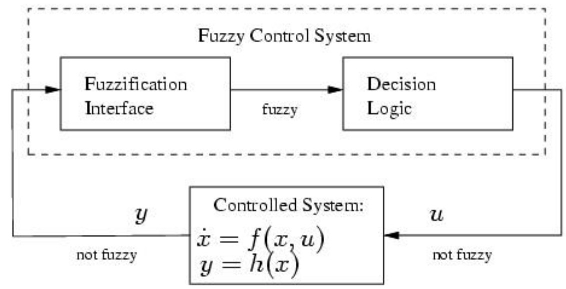


Fig. 1. Architecture of a Takagi and Sugeno fuzzy controller.

$\kappa_i(y)$ for $i = 1, \ldots, N$, whose members become applicable based on the classification of the output y of the system. The task of the decision logic is to determine the degree of applicability μ_i to each of the control laws based on fuzzy sets or membership functions, which are typically triangular or piecewise continuous functions. Of course, it is also possible to use other types of membership functions. The final control value is then computed as

$$u = \frac{\sum_{i=1}^{N} \mu_i(x) \kappa_i(x)}{\sum_{i=1}^{N} \mu_i(x)}. \tag{15}$$

The success of fuzzy control systems depends on a number of parameters including the fuzzy membership function, yet they are often selected subjectively and then tuned manually to improve performance. Some work has been done on tuning these parameters using genetic algorithms as well as other methods [20], [21], [22]. We will show that these fuzzy sets can be optimized using the variational approach presented earlier. This method relies on the dynamics of the controlled system, and requires that the state transition functions and the membership functions are continuously differentiable. In particular, we will use this technique to derive optimal membership functions for a particular robotics example.

V. ROBOTICS APPLICATION

In this section, we will derive a fuzzy logic controller with optimal membership functions to control a unicycle with two control laws that correspond to "go-to-goal" behavior and

”avoid-obstacles” behavior. The dynamics of the unicycle are

$$\begin{aligned} \dot{x} &= v\cos(\phi), \\ \dot{y} &= v\sin(\phi), \\ \dot{\phi} &= \omega. \end{aligned} \tag{16}$$

In the system above (x, y) is the Cartesian coordinate of the center of the unicycle and ϕ is its orientation with respect to the x-axis. We further assume that v is constant and ω is the control variable. The system has two behaviors, namely ”go-to-goal” and ”avoid-obstacle”, and the control laws associated with each behavior are as follows:

$$\omega_g = \kappa_g(x, y, \phi) = C_g(\phi_g - \phi), \tag{17}$$

$$\omega_o = \kappa_o(x, y, \phi) = C_o(\pi + \phi_o - \phi). \tag{18}$$

Note here that C_g and C_o are the gains associated with each behavior, and ϕ_g and ϕ_o are the angles to the goal and nearest obstacle respectively. Both of these angles are measured with respect to the x-axis and can be expressed as

$$\phi_g = \arctan(\frac{y_g - y}{x_g - x}) \text{ and } \phi_o = \arctan(\frac{y_o - y}{x_o - x}), \tag{19}$$

where (x_g, y_g) and (x_o, y_o) are the Cartesian coordinates of the goal and the nearest obstacle respectively. So for our fuzzy controller, the control w will be given by

$$w = \frac{\mu_g(x)\omega_g + \alpha_o(x)\mu_o}{\mu_g(x) + \mu_o(x)}. \tag{20}$$

The two control laws are weighted by μ_g and μ_o computed according to the corresponding membership function.

Now that the individual control laws are defined, we must next decide on the appropriate membership functions which determine the degree of applicability of each law. This is typically done by using piecewise linear or triangular membership functions as shown in Figure 2, but we need these functions to be continuously differentiable. Thus we will specify them using exponential functions (e.g. $e^{-\alpha\|x-x_o\|^2}$), and optimize them with respect to the tuning parameter α. In particular, the membership functions are defined as

$$\mu_g(x, \alpha_g) = 1 - e^{-\alpha_g\|x-x_o\|^2}, \tag{21}$$

$$\mu_o(x, \alpha_o) = e^{-\alpha_o\|x-x_o\|^2}. \tag{22}$$

Moreover, we let

$$L(x(t), z(t)) = ae^{-b\|z-z_0\|^2} + c \parallel z - z_g \parallel^2, \tag{23}$$

where $z = [x, y, \phi]^T$ and $\dot{z}$ is given by (16) and (20). Also let $\psi(x(T), z(T)) = 0$ since the control (20) ensures that the unicycle will reach the goal given that the goal is sufficiently far away from an obstacle. The result of the gradient descent algorithm and the optimal trajectory are shown in Figure 3. For the simulation, $z_0 = [-1.5, 0, 0]^T$, $z_o = [0, 0]^T$, $z_g = [3, 0]^T$, $a = 2$, $b = 10$, and $c = 0.01$. The results show that $\alpha_g^* = 5.5584$ and $\alpha_o^* = 1.2068$.

With the optimal membership functions designed for this known environment, the resulting fuzzy logic control strategy can easily be transitioned onto a real robotic platform navigating in an unknown environment. Note that since the optimization is performed over a well-defined environment, the resulting navigation strategy will no longer be optimal in an unknown environment but rather corresponds to a suboptimal performance enhancing strategy. To illustrate this point, we compare the performance of the optimal fuzzy controller to the standard fuzzy controller with triangular functions on the Magellan Pro platform from iRobot with the setup shown in Figure 4. The resulting trajectories are plotted using the odometry and sensor readings from the robot. The standard fuzzy controller resulted in a trajectory with cost $J_{std} = 8.8603$, while the optimized fuzzy controller lowered the cost to $J_{opt} = 7.7641$. It should moreover be noted that both of these controllers outperformed the standard hybrid controller where the system switches between the control laws according to the distance from the nearest obstacle. The switched system resulted in a trajectory with cost $J_{switch} = 9.3674$. The results show that the approach presented in this paper offers a novel systematic approach for fine-tuning fuzzy controllers by optimizing the corresponding membership functions, thus resulting in improved performance.

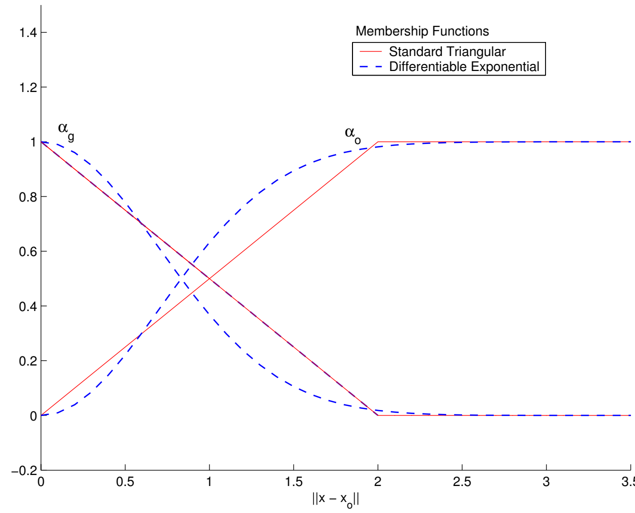
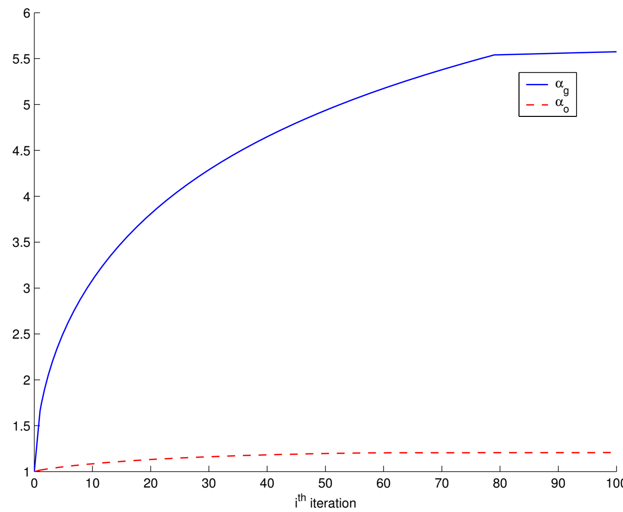


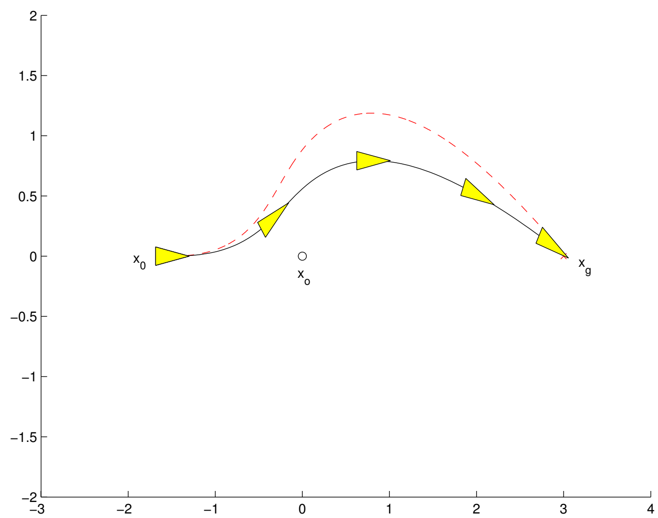
Fig. 2. The standard piecewise linear or triangular (solid) and the differentiable exponential (dashed) membership functions.

VI. CONCLUSIONS

When humans acquire new motor skills, they are typically obtained from a combination of previously established skills. Based on this observation, we proposed a systematic approach for a developing new controller as an optimal fusion of existing controllers. The method employs variational arguments to optimize parameterized membership functions in a general settings. This view complements the standard fuzzy logic control approach, where the modes are combined according to some given (and possibly fine-tuned) membership functions. A number real robotics tests were conducted to verify the performance benefit of the proposed method, as well as its numerical stability and computational viability.

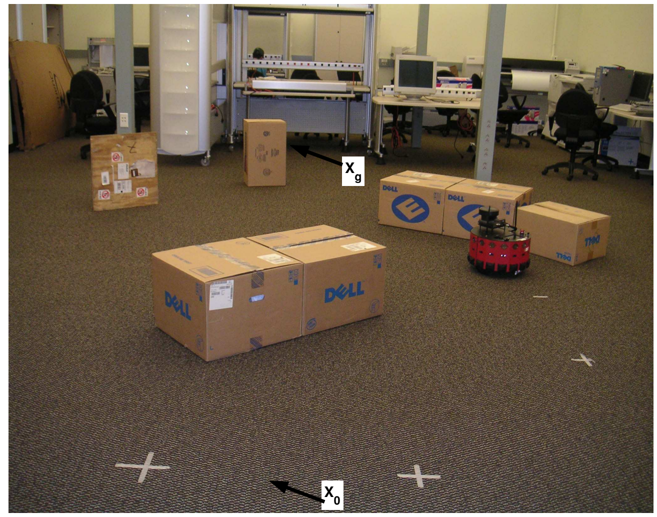


(a)

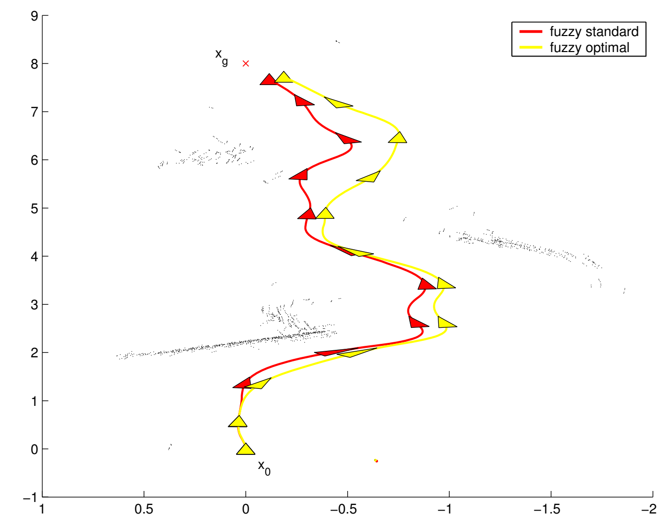


(b)

Fig. 3. Optimization results: (a) The evolution of α_g and α_0, (b) The trajectory for the initial guess of α (dashed) along with the final optimal trajectory.



(a)



(b)

Fig. 4. (a) The experimental setup, (b) The resulting trajectories plotted using the odometry readings, while the obstacle are inferred from the sensor readings.

REFERENCES

[1] T. A. Henzinger. The theory of hybrid automata. *Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS)*, IEEE Computer Society Press, 1996, pp. 278-292.

[2] D. Hristu-Varsakelis, M. Egerstedt, and P.S. Krishnaprasad. On The Structural Complexity of the Motion Description Language MDLe. *IEEE Conference on Decision and Control*, Dec. 2003.

[3] M. Egerstedt. On the Specification Complexity of Linguistic Control Procedures. *International Journal of Hybrid Systems*, Vol. 2, No. 1-2, pp. 129-140, March & June, 2002.

[4] M. Sugeno. An Introductory Survey of Fuzzy Control. *Information Sciences*, 36:59-83, 1985.

[5] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*, Springer, Berlin, 1993.

[6] R. Kruse, J. Gebhardt, F. Klawonn. *Foundations of Fuzzy Systems*, John Wiley & Sons, 1994.

[7] G. J. Pappas. Bisimilar linear systems. *Automatica*, 39(12):2035-2047, December 2003.

[8] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences* 57:94–124, 1998.

[9] A. Bicchi, A. Marigo, and B. Piccoli. On the reachability of quantized control systems. *IEEE Transactions on Automatic Control*, 4(47):546-563, April 2002.

[10] E. Asarin, O. Bournez, T. Dang, O. Maler. Reachability Analysis of Piecewise-Linear Dynamical Systems *Hybrid Systems: Computation and Control*, 20-31 LNCS 1790, Springer, 2000

[11] I. Mitchell and C. J. Tomlin. Overapproximating Reachable Sets by Hamilton-Jacobi Projections *Journal of Scientific Computation*, Volume 19, Number 1, pp. 323-346, December 2003.

[12] A. Bhatia, and E. Frazzoli. Incremental Search Methods for Reachability Analysis of Continuous and Hybrid Systems. *Hybrid Systems: Computation and Control. Springer-Verlag*, 2004.

[13] T. Mehta and M. Egerstedt. Learning Multi-Modal Control Programs. *Hybrid Systems: Computation and Control, Springer-Verlag*, March 2005.

[14] P. Cheng, Z. Shen, and S. M. LaValle. Using randomization to find and optimize feasible trajectories for nonlinear systems. *Proc. Annual Allerton Conference on Communications, Control, Computing*, pages 926–935, 2000.

[15] S. M. LaValle. From dynamic programming to RRTs: Algorithmic design of feasible trajectories. *Control Problems in Robotics*, pages 19–37. Springer-Verlag, 2002.

[16] L.P. Kaebling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal Of Artificial Intelligence Research*, 1996.
[17] C.J.C.H. Watkins, and P. Dayan. Q-learning. *Machine Learning* 8(3/4):257-277, May 1992.
[18] T. Mehta and M. Egerstedt. Motion Alphabet Augmentation Based on Past Experience. *IEEE Conference on Decision and Control*, December 2005.
[19] L. Armijo. Minimization of Functions Having Lipschitz Continuous First-Partial Derivatives. *Pacific Journal of Mathematics*, Vol. 16, ppm. 1-3, 1966.
[20] D. Park, A. Kandel, and G. Langholz. Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control. *IEEE Transactions on Systems, Man and Cybernetics*, Volume 24, Number 1, January 1994.
[21] S.C. Lin; Y. Chen. Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control. *IEEE Transactions on Evolutionary Computation*, Volume 2, pages: 846 - 851, 1995
[22] K. Tanaka, T. Hori, H.O. Wang. New Robust and Optimal Designs for Takagi-Sugeno Fuzzy Control Systems. *IEEE International Conference on Control Applications*, Volume 1, pages 415-420, 1999.