Journal of Computer Science 6 (10): 1019-1026, 2010
ISSN 1549-3636

Adaptive Fuzzy Knowledge Based Controller for Autonomous Robot Motion Control

Mbaitiga Zacharie
Department of Media Information Engineering,
Okinawa National College of Technology,
905 Henoko, Nago, 905-2192, Okinawa, Japan

Abstract: Problem statement: Research into robot motion control offers research opportunities that will change scientists and engineers for year to come. Autonomous robots are increasingly evident in many aspects of industry and everyday life and a robust robot motion control can be used for homeland security and many consumer applications. This study discussed the adaptive fuzzy knowledge based controller for robot motion control in indoor and outdoor environment. **Approach:** The proposed method consisted of two components: the process monitor that detects changes in the process characteristics and the adaptation mechanism that used information passed to it by the process monitor to update the controller parameters. **Results:** Experimental evaluation had been done in both indoor and outdoor environment where the robot communicates with the base station through its Wireless fidelity antenna and the performance monitor used a set of five performance criteria to access the fuzzy knowledge based controller. **Conclusion:** The proposed method had been found to be robust.

Key words: Adaptive fuzzy, fuzzy control, fuzzy knowledge, motion control

INTRODUCTION

Autonomous robots are intelligent machines capable of performing tasks by themselves without human control over their movements. These intelligent machines have proliferated in recent years and promise to play a major role in our lives in the future. In artificial intelligent robot are known as “agent”, they are distinguished from software agents in the way they are embodied and situated in the real world. They received information from the world through their sensors. They can be touched and seen and heard, they have physical dimensions and they can exert forces on other objects. These objects can be balls to be kicked, parts to be assembled, terrain to be traversed or carpets to be vacuumed. Robots are also subjects to the world’s physical laws, they have mass and inertia, their moving parts encounter friction and hence produce heat, not two parts are precisely alike, measurements are corrupted by noise and parts break. Robots contain computers, which provide them with ever-increasing speed and power for both signal processing and cognitive functions. The world into which we place these robots keeps changing; it is non stationary and unstructured, so we cannot predict their behavior accurately in advance. These are some of the features of the autonomous robots. They suffer from all the limitations of the real world but because they are physical they also fascinate us. This is particularly true for humanoid robots, but there is some intrigue in all moving robots. They are an imitation of the life and we are drawn to watching them. It is not only the fact that they move, that beguiles us, since many things move in the world, but they appear to move intelligently, they avoid obstacle, they interact with one another and they accomplish tasks. For those of us who design and build them, enabling them to perform these and others actions is precisely our goal. But what is robot control in all senses? To this question, there appears to be a contradiction between autonomy which implies that a robot is capable of taking care of itself and control which appears to imply some sort of human intervention. To be sure some form of high-level control is required to ensure that the robots do not harm any humans, equipments or other robots. In effect, this high-level of control implies the implementation of Asimov’s laws, which can be paraphrased as follows:

- A robot should never harm human being
- A robot should obey a human being unless this contradicts the first law
- A robot should not harm another robot, unless this contradicts the first and second law

Most of the real-world process that requires automatic control are nonlinear in nature. That is to say, their parameter values alter as the operating point changes over time. As conventional control schemes are linear, a controller can only be tuned to give good performance at a particular operating point or for a limited period of time. The controller needs to be retuned if the operating point changes, or retuned periodically if the process changes with time. This necessity to retune has driven the need for adaptive controller that can automatically retune themselves to match the current process characteristics. An excellent introduction to conventional adaptive control systems can be found by (Aström and Witternmark, 1994) Fuzzy Knowledge Based Controller (FKBC) is nonlinear and so they can designed to cope with a certain amount of process linearity. However, such design is difficult, especially if the controller must cope with nonlinearity over a significant portion of the operating range of the process. Also, the rules of the FKBC do not in general contain a temporal component and they cannot cope with process changes over time. So there is a need for Adaptive Fuzzy Knowledge Based Controller (AFKBC). The adaptive component of the AFKBC we designed consists of two parts:

- The first part is the performance monitor that detects changes in the process characteristics by assessment of the controlled response of the process
- The second part is the adaptation mechanism. It uses information passed to it by the performance monitor to update the controller parameters and so adapts the controller to the changing process characteristics

MATERIALS AND METHODS

Robot mechanics: The robot shown in Fig. 1 has a wheel-based platform of two 12VDC motors each supply 300 inches (22 kg) of torque to the X80's 7-inches (18 cm) wheels, yielding a top speed in excess of 1 m sec^{-1} (3,3 ft sec^{-1}).

Two high resolution of 1200 counter per wheel cycle quadrature encoders mounted on each wheel provide high-precision measurement and control of wheel movement. Weighting only 7.7 lb, the system is light, but it can carry an additional payload of 22 lb (10 kg).

Robot sensors: The robot offers full Wi-Fi 802.11 g wireless, multimedia, sensing and motion capabilities and comes with a wide range of sensor, camera and audio modules, sufficient to serve in any variety of applications. The robot offers broad expandability as well for projects that may require additional sensors, even specialized modules. Powered by separate RC servo motors, the integrated camera head can pan and tilt independently. With its integrated high bandwidth of 11 Mbps, Wi-Fi 802.11 g wireless module, the system can upload all sensor data including encoder sensor readings to a PC or server at rates in excess of 10 Hz. Similarly, streaming audio (8 Hz × 8 bits) and video (up to 4 fps) either for direct monitoring or for processing by high-level Artificial Intelligent (AI) schemes is a snap. Commands and instructions sent to the robot via the same wireless link also pass at rates exceeding 10 Hz, providing real-time control and access. The robot includes all Wi-Robot development software components, enabling easy access to all data and information in a standard Microsoft Windows programming environment example, Visual Basic and Visual Studio C++. Under the approach of using a separate PC for high-level control, there are no longer onboard restrictions on a mobile system's processing power, memory and storage.

Robot kinematics: In our proposed AFKBC, we considered the problem of coordinating the position and orientation of the robot instead of coordinating its center position. We defined the position of the robot at point P = [P_x, P_y] that lies a distance L along the line that is normal to the wheel axis and intersects the wheel axis at the center point r = [r_x, r_y] as shown in Fig. 2.

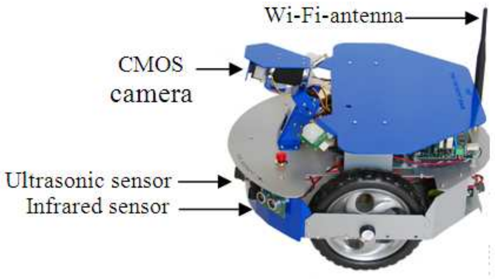


Fig.1: X80 Robot model used for this study

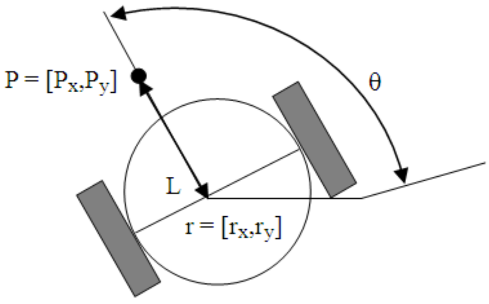


Fig. 2: Nonholonomic differentially driven wheeled robot

The kinematic of the robot position are nonholonomic for L ≠ 0. Now Let (r_x, r_y), θ and (v, ω) denote the inertial position, orientation and linear and angular speeds of the robot respectively. The kinematic equations for the robot are:

$$\dot{r}_x = v_x \cos\theta, \dot{r}_y = v_y \sin\theta, \dot{\theta} = \omega \qquad (1)$$

The position is given by:

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \end{bmatrix} + L \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad (2)$$

Differentiating (2) with respect to time gives:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (3)$$

Letting:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{L}\sin\theta & \frac{1}{L}\cos\theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \qquad (4)$$

Gives:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \qquad (5)$$

which is a simplified kinematic equation of our robot, but is sufficient for the purpose of our application. Note that (5) takes in the form of single-integrator dynamics, implying that the consensus algorithm (6) can be directly applied:

$$u_i = \dot{r}_i^d - \alpha_i (r_i - r_i^d) - \sum_{j=1}^{n} a_j^v \left[(r_i - r_i^d) - (r_j - r_j^d) \right] \qquad (6)$$

Where:

$r_i \in R$ = The state and $u_i \in R$ is the control output of the robot

α_I = A positive scalar

a_j^v = The (i,j) entry of n×n adjacency ν_n^v matrix associate with the interaction topology:

$$\Im_n^v = (\nu_n^v, \varepsilon_n^v) \text{for } r_i - r_i^d \text{ and } r_i^d = (x_i^d, y_i^d) \qquad (7)$$

With:

$$\begin{bmatrix} x_i^d \\ y_i^d \end{bmatrix} = \begin{bmatrix} x_{ci} \\ y_{ci} \end{bmatrix} + \begin{bmatrix} \cos\theta_{ci} & -\sin\theta_{ci} \\ \sin\theta_{ci} & \cos\theta_{ci} \end{bmatrix} \begin{bmatrix} x_{iF}^d \\ y_{iF}^d \end{bmatrix} \qquad (8)$$

Where:

$r_i^d = \left[x_i^d, y_i^d \right]$ = The robot actual and desired position

$r_{iF}^d = \begin{bmatrix} x_{iF}^d \\ y_{iF}^d \end{bmatrix}$ = The desired deviation of the robot relative to C_F

C_F = Denotes a virtual coordinate frame located at a virtual center (x_c, y_c) with an orientation θ_c relative to C_0

C_0 = The initial frame

Fuzzy control and related works: Fuzzy control has met a tremendous interest in application over the past few years. Precisely in manufacture control equipment and automatic control than traditional control techniques. There are a number of reasons for the popularity of fuzzy control. First, fuzzy control is a real-time expert system, implementing a part of human operator’s or process engineer’s expertise which does not lend itself to being easily expressed in PID parameters or differential equations but rather a situation or action rules. Second, fuzzy control offers a better user interface to the process to be controlled by translating the system insight into controller nonlinearities. Nonlinear characteristics are realized in fuzzy control by partitioning the rule space, weighting the rules and by nonlinear membership function. Rule-based systems compute their output by combining results from different parts of the partition, each part being governed by separate rules. In fuzzy reasoning, the boundaries of these parts overlap and the local results are combined by weighting them appropriately. That is why the output in fuzzy system is a smooth, nonlinear function. However, fuzzy control differs from main-stream expert system technology in several aspects. One main feature of fuzzy systems is their existence at two distinct levels: First, there are symbolic if-then rules and qualitative, fuzzy variable and values such as if pressure is high and slightly increasing then energy supply is medium negative. These values such as “slightly increasing” and fuzzy operators such as “and” are compiled into numerical objects and algorithms: Function tables, interpolation and comparators. Due to its easy use and its better performance; fuzzy logic control has become the most popular control tool for many researchers in the field of robotics control (Cardenas, 2004; Elmaliach *et al*., 2007; Martins-Filho and Macau, 2007; Ryu *et al*., 2006; Peri, 2002; Cho and Youn, 2006; Nakamura and Sekiguchi, 2001). For example in (Astudillo *et al*., 2006) the authors proposed an intelligent control of an autonomous mobile robot using Typ-2-Fuzzy Logic. In their method, the fuzzy control has two input and two output variables and the membership functions are defined by one triangular and trapezoidal function for each variable involved due to the fact that they are easy to implement computationally. The membership functions in which Negative, Zero and Positive representing the fuzzy set are associated to each input and output variable, where the universe of discourse is normalized into [-1,1] range. The rule set of the Fuzzy Logic Control (FLC) contains 9 rules which governing the input-output relationship of the FLC and this adopts the Madami-style engine (Passion and Yurkovich, 1998). Jin (1998) a decentralized adaptive fuzzy control of robot manipulators has been proposed having two input and two output variables. In the fuzzy inference system, the membership functions of the linguistic variables are sum triangular functions. Both input variables in each link are partitioned into four fuzzy subsets and 16 fuzzy rules are set up for each link. Then genetic algorithm is used to tune the parameters so that the fuzzy system can realize the mapping of the inverse robot dynamics. The population of the genetic algorithm is 50 and the length of the gene is 160. But what is very interesting in this proposed method is that the premise variable parts of the adaptive fuzzy logic in the consequence parts of the rules have been not used. In spite of not using the fuzzy logic premise the performance of the controller has some amazing point in the simulation results. Takagi and Seguna (1994) have used the same method and they suggested that if the rule has M rules together, the final output of the fuzzy model is calculated as follows:

$$\tau(k) = \frac{\sum_{i=1}^{M}\{w^i(k)\tau_0{}^i\}}{\sum_{i=1}^{M} w^i(k)} \quad (9)$$

With:

$$w^i(k) = \min\{A_1^i(q^d(k)), A_2^i(\dot{q}^d(k))\} \quad (10)$$

Where:

A_1^i and A_2^i = The fuzzy sets for q^d and $\dot{q}^d$
τ_0^i = The crisp output of each fuzzy rule
K = The time constant

Given a set of input-output data, the premise and consequence parameter can be determined by the used of complex search algorithm and recursive least-square algorithm (Sugeno and Tanaka, 1994). Tunstel *et al*. (2002) proposed the combination of fuzzy Logic controller and Fuzzy Knowledge-Based Decision system for multi robot control with single input and output.

AFKB control approach: Parameter estimator: In the parameter estimator, changes in process are detected through off-line identification of the process model. A process model here is referred to a mathematical description of the process that gives values of the process-output, given the current process-state and the input to the process. For space constraints the process model has not been described. The process input come from the controller. The model used in the Single-Input and Single-Output (SISO), linear, first order plus dead time is defined by the transfer function (11), which has been used in the self-tuning regulator.

$$\frac{y(x)}{u(x)} = \frac{Kpe^{-tds}}{\tau s + 1} \quad (11)$$

Where:

y(s) = The Laplace transform of the process-output
u(s) = The Laplace transform of the process-input
K_p = The gain
t_d = The dead time
τ = The time constant

The parameter estimator must continually estimate values for the process parameters, K_p, τ and t_d and should appropriately update them according to the robot position by taking into consideration the information coming from the obstacle avoidance sensors. The rule used for this AFKBC consists of a set of if-then rule of the form:

$$\text{Rulei} : \text{ifx is } \Im_i \text{then u is } \Re u^{(i)} \quad (12)$$

Where:

X = The controller inputs (process-state variable)
U = The controller output variable
I = The rule number (i =1, 2, 3, ….., N)
$\Im_I$ = The linguistic value of the rule-antecedent
$\Re_u^{(i)}$ = The linguistic value of the rule consequent

The membership functions $L\tilde{\Im}_j$ of the antecedent are triangles described by a peak value, a_{ij} and a support, b_{ij}, in the universe of discourse as shown in Fig. 3.

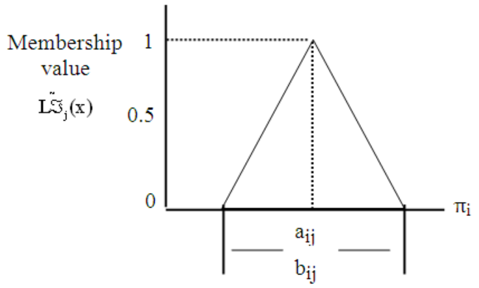


Fig. 3: Triangular AFKBC fuzzy set

The membership function of the AFKBC is:

$$L\tilde{\Im}_j^{(i)}(x) = 1 - \frac{|x - a_{ij}|}{b_{ij}} \quad (13)$$

Where:

X = The crisp controller input in the domain π_i of the input variable x_j. The control-output membership function

$\tilde{\Re u}^{(i)}$ = A fuzzy singleton set defined on the real number $u_{i.}$

That is to say:

$$\tilde{\Re u}^{(i)} = \frac{1}{u_i} \quad (14)$$

Using the max-product composition and the Center-of-Area defuzzyfication method the fuzzy output u* from the rule set has been obtained as follows:

$$u^* = \frac{\sum_{i=1}^{n} \mu_i \cdot u_i}{\sum_{i=1}^{n} \mu_i} \quad (15)$$

where, μ_I is Given by the product of the membership degrees of the input variables in the i-th rule:

$$L\tilde{\Im}_1^{(i)}(x_1) \cdot L\tilde{\Im}_2^{(i)}(x_2) \cdot L\tilde{\Im}_3^{(i)}(x_3) \ldots\ldots\ldots L\tilde{\Im}_N^{(i)}(x_N) \quad (16)$$

where, x_j is a crisp controller input variable from the domain π_i of the input variable $\Im_i$. Figure 4 shows the graphical representation of the Center-of-Area defuzzyfication.

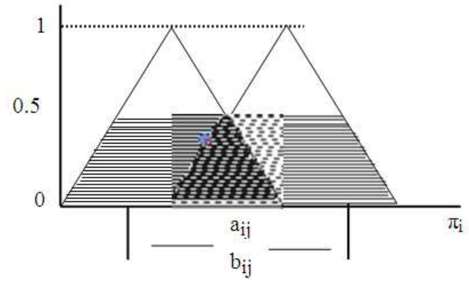


Fig. 4: Graphical representation of the Center-of-Area defuzzication

Table 1: Control output

Large decrease	No change	Large increase
Small decrease		Small increase

Tuning procedure: The AFKBC rules are extracted from process operators. The membership functions are defined initially such that the input domains are divided equally by a suitable choice of the a_{ij} and the sets overlap, by a suitable choice of $b_{ij.}$ The u_i are chosen to give a suitable range of control-output covering a large decrease, small decrease, no change, large increase, small increase as shown in Table 1.

After the set of reliable controller input-output data (u^r) has been collected, the tuning procedure has been carried out as follows:

- The rules are fired on the input data to obtain the antecedent value, μ_i, for each rule and the real-value control-output u
- Parameters u_i are updated using (17):

$$u_i(t+1) = u_i(t) - \frac{K \cdot \mu_i}{\sum_{K=1}^{n} u_i} \cdot (u - u^r) \quad (17)$$

where, K is a constant which controls how much the parameters are altered at each iteration. As the iterations proceed, the objective function converges to a local minimum. In our AFKBC the objective function parameters we wish to alter are the membership functions $a_{ij,}$ b_{ij} and u_i

- Rule firing is repeated using the new values of u_i
- Parameters a_{ij} and b_{ij} are updated using the new values of u_i, μ_I and u respectively
- Inference error $D(t) = \frac{1}{2(u(t) - u^r)^2}$ is calculated
- If the change-of-error $|D(t) - D(t-1)|$ is suitable small, the tuning procedure is completed; otherwise it is repeated from Step 1

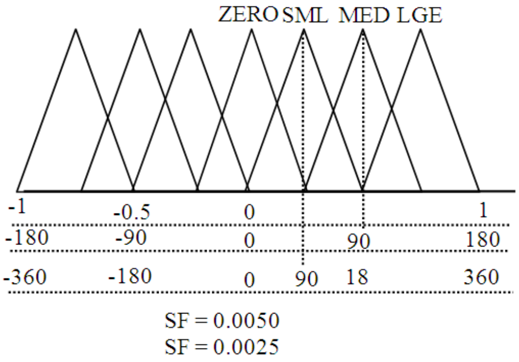


Fig. 5: The effect of altering scaling factor

This procedure will modify the actual values of u_i used for the controller output and will change the center, a_{ij} and width b_{ij}, of the antecedent fuzzy set.

Adaption mechanism: The adaptation mechanism modified the controller parameter in order to improve the controller performance on the basis of the output from the process monitor. We classified the adaptation mechanism for AFKBC according to which parameters are adjusted. Parameters are adjusted including the scaling factors with which controller input and output values are mapped onto the universe of discourse of the fuzzy set definitions. The fuzzy set a definition is defined in a normalized universe from -1.0 to +1.0 range as illustrated in Fig. 5.

Where SML is small, MED is medium and LGE is large and SF is a scaling factor. The real values of this input variables is ranged from -180 to +180 and so need to be scaled. If for example, the input value is multiplied by a scaling factor of 0.0050, the input is mapped to the universe of discourse as shown by the middle scale of Fig. 5. In this case an input value of 90 is classified as MEDIUM. Altering the scaling factor changes the classification of an input value. For example, with a scaling factor of 0.0025 a value of 90 is now classified as SMALL as shown in the bottom of Fig. 5. This reduces the sensitivity of the controller to the input and so reduces the controller gain.

RESULTS

In order to check the performance of our controller we validated the proposed Adaptive Fuzzy Knowledge-Based Controller on single experimental robot platform. The robot communicates with the base station through its Wireless fidelity antenna fixed on the robot back. The commands and instructions are sent to the robot from the base station.

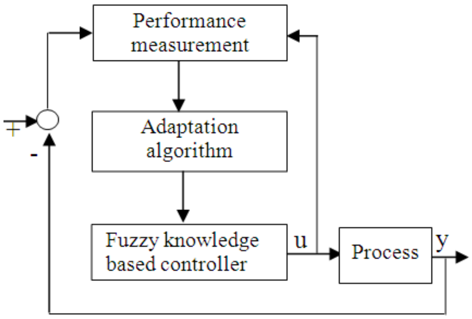


Fig. 6: Adaptive controller structure

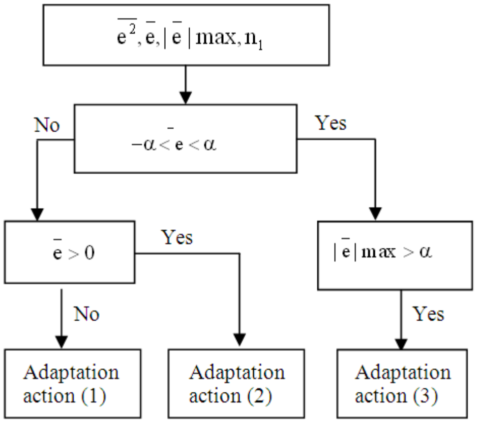
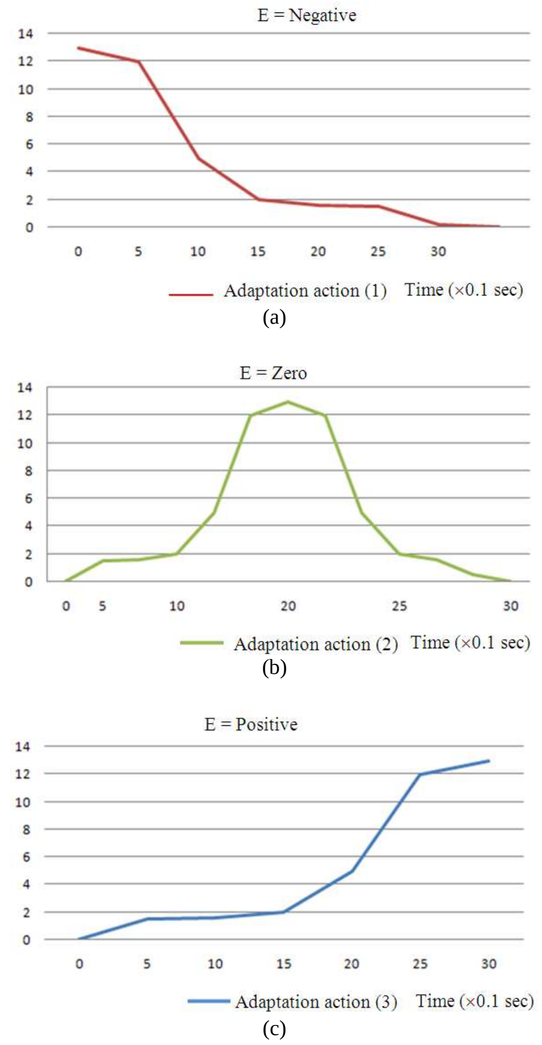


Fig. 7: Adaptive action algorithm

Our performance monitor used a set of five performance criteria to access the Fuzzy Knowledge Based Controller. These five criteria are:

- Average square error $\overline{e^2}$
- Average error $\bar{e}$
- Average absolute error $|\bar{e}|$
- Maximum absolute error $|\bar{e}|\max$
- Number n_1 of consecutive variations in control output

These five performance criteria are evaluated over a fixed observation period, the length of which is a tuning parameter for the controller. As the primary aim of our proposed controller is to maintain the process at set-point the first four performances, criteria are of major importance. The last criteria express the number of the secondary control objective of reducing the number of command variations. Figure 6 shows the structure of the controller and the adaptation algorithm concerned with improving the set-point control is shown in Fig. 7 respectively.



(a)

(b)

(c)

Fig. 8: Performance response of adaptation action

where, α is the membership function threshold. The three different adaptations are taken depending on the value of the four performance criteria. If the average error is too large, then adaptation Action (1) or (2) is carried out depending on whether the average error is below or above the set point, respectively. An intermediary adaptation is taken if the average squared error is too large indicating imprecision controḷ. Finally adaptation Action (3) is taken if the error becomes too large even for a sampling instant.

All the adaptation actions result in changes to fuzzy set membership definitions for error, change-of-error and change in control output. The adaptation (1) is concerned with improving the controller performance when the process is consistently below set-point. This has been done by increasing the degree to which values of the error are recognized as negative, that is to say, below the set-point. The performance response of Action (1-3) are illustrated in Fig. 8 where E denotes the error. As the aim of our adaptive process is to provide quick controller adaptation without causing instability or oscillations, only small changes have been made to the fuzzy set definition at any time. In addition the adaptation process needs not to be carried out at every sampling time. Because adapting for set-point aggressively will lead to instability.

DISCUSSION

In everyday language to adapt means to change behavior to conform to a new circumstance. Intuitively, an adaptive controller is thus a controller that can modify its behavior in response to changes in the dynamics of the process and the character of the disturbances. In other word a controller with adjustable parameters and a mechanism for adjusting the parameters. The controller becomes nonlinear because of the parameter adjustment.

In the proposed AFKBC the adaptation is done by modifying the membership function in proportion to the undesired effects that are being corrected. The threshold values have been carefully chosen as they can have a large effect on the overall control system performance. The parameters are concerned with specifying what levels of the set-point error criteria are acceptable before adaptation can carry out. If these parameters are set too low, the controller will constantly trying to adapt and may never be able to satisfy the required performance we would like to obtain. If these parameters are set too high, the controller may give unnecessarily bad performance as a little adaptation will be carried out. So we set the parameters for our controller in the median position that is to say between the high and low level where an excellent performance is obtained as the adaptation mechanism adapts the controller to the changing process characteristics. (Nomura *et al*., 1991) have applied the adaptive self-tuning method for their mobile robot avoiding a moving obstacle with some success but comparing to our controller, their controller need to be improved as its shown some instabilities. The robot problem was for the robot to move from its starting point to a robot fixed target point while avoiding a single obstacle moving across its path. Only the steering angle of the robot was controlled, while its driving speed was constant. The dynamics of the robot were not considered, while in our proposed AFKBC we take into consideration the robot dynamics.

The robot received four inputs. Five membership functions were defined for each of the four inputs and an initial set of the 625 rules was provided. Bartolini *et al.* (1982) have chosen to monitor the adaptation procedure for excessive commands variations at every sampling time so that this could be quickly corrected, in their method: Development of performance adaptive fuzzy controller with application to continuous casting plant. But they monitored set-point control performance at a lower rate.

Today, a large number of industrial controls are under adaptive control. These include a wide range of applications in aerospace process control, ship steering, robotics automotive and biomedical systems. The applications have shown that there are many cases in which adaptive control is very useful, others in which the benefits are marginal and yet others in which it is inappropriate. On the basis of the products and their uses, it is clear that adaptive control techniques can be used in many different ways.

CONCLUSION

In this study, the adaptive fuzzy knowledge based controller for robot motion control is proposed. The adaption component of the AFKBC is composed of two parts: the parameter estimator that constantly updates the process and the adaptation mechanism alters the controller parameter estimator on the basis of any detected changes. To access to the Fuzzy Knowledge Based Controller the parameter estimator used five performance criteria and maintain the process at controller set-point. Commands and instructions sent to the robot via its wireless link pass at rates exceeding 10Hz, providing real-time and control access. The evaluation test has demonstrated that the proposed controller study quite well, comparing to previous study in adaptive fuzzy logic or fuzzy logic control.

REFERENCES

Aström, K.J. and B. Witternmark, 1994. Adaptive Control. 2nd Edn., Prentice Hall, Boston, MA., USA., ISBN: 0201558661, pp: 580.

Astudillo, L., O. Castillo, P. Melin, A. Alanis and J. Soria *et al*., 2006. Intelligent control of an autonomous mobile robot using Type-2 fuzzy logic. Eng. Lett., 13: 1-5.

Bartolini, G., G. Casalino, F. Davoli, M. Masttreta and R. Minciardi *et al*., 1982. Development of Performance Adaptive Fuzzy Controller with Application to Continuous Casting Plant. In: Cybernetics and Systems Research, Trappl, R. (Ed.). North-Holland, Amsterdam, pp: 721-728.

Cardenas, A.M., 2004. Intelligent car parking using fuzzy-neural networks. Proceedings of the 2nd LACCEI International Latin American and Caribbean Conference for Engineering and Technology, June 2-4, UPC, Miami, Florida, USA., pp: 1-6.

Cho, Y. and J.H. Youn, 2006. Wireless sensor-driven intelligent navigation robots for indoor construction site security and safety. Proceedings of the 23rd International Symposium on Automation and Robotics in Construction, Oct. 3-5, IAARC, Tokyo, Japan, pp: 493-498.

Elmaliach, Y., N. Agmon and G.A. Kaminka, 2007. Multi-robot area patrol under frequency constraints. Proceedings of the IEEE International Conference on Robotics and Automation, Apr. 10-14, IEEE Xplore Press, Roma, pp: 385-390. DOI: 10.1109/ROBOT.2007.363817

Jin, Y., 1998. Decentralized adaptive fuzzy control of robot manipulators. IEEE Trans. Syst. Man Cybernet. B Cybernet., 28: 47-57. PMID: 18255921

Martins-Filho, L.S. and E.E.N. Macau, 2007. Patrol mobile robots and chaotic trajectories. Math. Probl. Eng., 2007: 1-13. DOI: 10.1155/2007/61543

Nakamura, Y. and A. Sekiguchi, 2001. The chaotic mobile robot. IEEE Trans. Robot. Autom., 17: 890-904.

Nomura, H., I. Hayashi and N. Wakami, 1991. A self-tuning method of fuzzy control by descent method. Proceedings of the 4th International Farming System Association World Congress, (IFSA'91), IEEE Press, San Diego Brussels, pp: 155-158 .

Passion, K.M. and S. Yurkovich, 1998. Fuzzy Control. 1st Edn., Addison-Wesley, USA., ISBN: 0-201-18074-X, pp: 475.

Peri, V.M., 2002. Fuzzy logic controller for an autonomous mobile robot. Bachelor of Technology in Electrical and Electronics Engineering Thesis, Jawaharlal Nehru Technological University. http://academic.csuohio.edu/embedded/Publications/Thesis/Mohan%20Thesis.pdf

Ryu, J.G., H.M. Shin, S.K. Kil, E.H. Lee and H.H. Choi *et al*., 2006. Design and implementation of real-time security guard robot using CDMA networking. Proceedings of the 8th International Conference on Advanced Communication Technology, (ACT'06), Robotics Lab, Spain, pp: 1-6.

Sugeno, M. and K. Tanaka, 1994. Successive identification of a fuzzy model and its application to prediction of complex system. Fuzzy Syst. Sets, 42: 315-324. DOI: 10.1016/0165-0114(91)90110-C

Takagi, T. and M. Seguna, 1994. Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Syst. Man Cybernet. B Cybernet., 15: 116-132.

Tunstel, E., A.A.M. de Oliveira and S. Berman, 2002. Fuzzy behavior hierarchies for multi-robot control. Int. J. Intell. Syst., 17: 449-470. DOI: 10.1002/int.10032