An Application of Fuzzy Logic Reinforcement Iterative Learning Control to Balance a Wheelchair

Phichitphon Chotikunnan [1], **Benjamas Panomruttanarug** [2], **Nuntachai Thongpance** [1], **Manas Sangworasil** [1], and **Takenobu Matsuura** [1],

ABSTRACT

This paper presents a novel control strategy to balance wheelchair using fuzzy logic reinforcement iterative learning to lift the front wheel and maintain system stability in the upright position based on fuzzy logic type-1 and type-2. With the iterative learning, the best set point of the system to be controled can be determined. Simulation results show the effectiveness of our method.

Keywords: Fuzzy Logic; Iterative Learning Control ;Wheelchair

1. INTRODUCTION

The iBOT is an electric powered wheelchair developed by Dean Kamen and other engineers of DEKA research & development corporation in the 1990s. It is designed to provide not only mobility for disabled users, but also several unique advantages over a traditional electric powered wheelchair. One of the significant advantages is that disabled people can use the iBOT to lift themselves up and to reach certain heights in confined spaces, for example, to put things on shelves, or to have conversations with other people at eye-to-eye level.

In order to raise wheelchair up to reach higher levels, the wheelchair has to be on two-wheels. This can be achieved by lifting up the front wheels (casters) of the wheelchair to be elevated to an upright position. When the wheelchair is on two wheels, it performs as a double inverted pendulum, therefore, it is characterized as a highly non-linear complex unstable system. For this reason, suitable controls are required to lift and stabilize the two-wheeled wheelchair in the upright position.

In our previous work, Self-Balancing iBOT-Like Wheelchair [3] used to Type-1 and Interval Type-2 Fuzzy Control for the inverted pendulum system has been proposed. A stabilization of the system was accomplished through computer simulation and experimental iBOT-Like Wheelchair [4] by test on carpet floor cement, floor and tile flooring and iBOT-Like Wheelchair to obtain self-balancing.

[1] Faculty of Biomedical Engineering, Rangsit University, Bangkok Thailand, E-mail: Phichitphon.c@rsu.ac.th

[2] Department of Control Systems and Instrumentation Engineering, King Mongkuts University of Technology Thonburi, Bangkok Thailand

Iterative learning control [5] (ILC) has been applied to systems performing a tracking maneuver repeatedly. The error in each run is used to adjust the command in the next run, aiming to converge to zero error. Control applications include high precision pick-and-place robots executing the same trajectory in every run. Case studies, robustification of iterative learning control and repetitive control are performed using a recently developed averaging technique to control robot [6] and this averaging technique is thus considered as most suitable for repetitive processes such as manufacturing [7-17].

In those methods have disadvantages that it is not easy to control the system since the parameters such as matrix of control gain and so on are fixed.Those make the system unstable.

In this paper in order to improve those disadvantages we propose a new method which is easy to control the system. Section 2 will describe the wheelchair mechanical design, lifting mechanism, and mathematical model in a state space representation. Section 3 will present fuzzy logic controllers design to stabilize a wheelchair on two wheels. Section 4 will present iterative learning control and Section 5 will describe adaptive iterative learning control with fuzzy logic control for automatic gain control. In Section 6, simulation results have also been conducted to show the effectiveness of the proposed control algorithm. Finally, the conclusions will be presented in Section 7.

2. THE WHEELCHAIR

2.1 Wheelchair structure

Figure 1 shows a wheelchair structure created by using computer-aided design software (Catia). The wheelchair has three degrees of freedom, e.g., forward/backward, pitch, and yaws, driven by eight DC motors. The first four motors are used to drive the left and right rear wheels to control motion (linear and steering). The next two motors are used to lift the front wheels (casters) to an upright position. In order to achieve the upright position, it is required to obtain certain amounts of torque to lift the front wheels. The last two motors are connected to arm robot for upper and lower altitude of robot.

This section presents the simulation results using the parameters of the dynamic model, as provided in Table 1.

Table 1:: Physical parameters for simulation results

Parameter	Value
m_1	3.2 kg
m_2	36.176 kg
l_1	0.145 m
l_2	0.4025 m
J_1	0.025024
J_2	1.73634
g	9.81 m/s^2

2.2 From a sitting to standing position

In order to drive on two-wheels, the wheelchair is initially set to four-wheels function. The arm robot is then slided from backward to the upper altitude of the robot, and then the COG is pushed toward to the rear wheels. When the COG is located over the rear wheels, the balancing process begins. Figure 2 presents the four transition phases to change the wheelchair's COG.

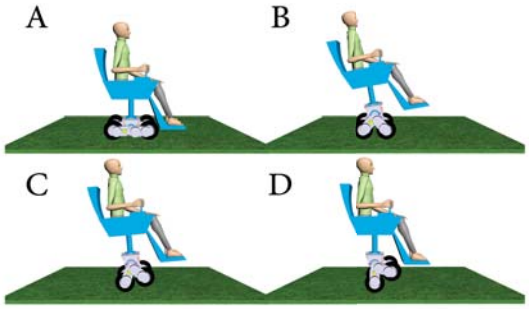

Fig.1:: The four transition phases for changing the COG: (a) four-wheel function; (b) COG is moving backward; (c) transition phase; and (d) balancing.

2.3 Dynamic Model of a two-wheeled wheelchair

In this section, we describe a mathematical model of the wheelchair to remain upright on two wheels. The wheelchair is considered to be a single link inverted pendulum. The non-linear dynamic model is analyzed based on derivation via the Euler-Lagrange equation of motion under the assumption that there is no slippage between the wheels and the ground. The details of the parameters used in the equations are defined in Appendix.

In order to obtain the wheelchair model, let us consider the system in two separate links, which are comprised of Link 1 and Link 2, as illustrated in Figure 3. Link 1 is composed of the rear wheels, whereas the front wheels and payload are considered as Link 2. One can write the kinetic energy of Link 1 as follows:

$$K_1 = \frac{1}{2}m_1\dot{Z}_1^2 + \frac{1}{2}J_1\frac{\dot{Z}_1^2}{l_1} \tag{1}$$

Similarly, the kinetic energy of Link 2 can be expressed as:

$$K_2 = \frac{1}{2}m_2(\dot{Z}_1^2 + 2\dot{Z}_1 l_2\dot{\theta}_2\cos\theta_2 + l_2^2\dot{\theta}_2^2) + \frac{1}{2}J_2\dot{\theta}_2^2 \tag{2}$$

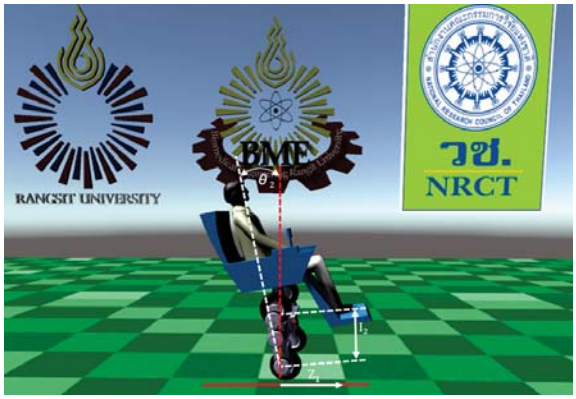

Fig.2:: Two-link wheelchair

Eq. (1) and Eq. (2) represent the total kinetic energy, K_T in the system. Since the potential energy of Link 1 is zero, the total potential energy, P_T, is only derived from Link 2, or $P_T = m_2 g l_2\cos\theta_2$. The Lagrangian function is therefore stated as follows:

$$\begin{aligned} L &= K_T - P_T \\ &= \frac{1}{2}(m_1 + m_2 + J_1)\dot{Z}_1^2 + \frac{1}{2}(m_2 l_2^2 + J_2)\dot{\theta}_2^2 \\ &\quad + m_2\dot{Z}_1\dot{\theta}_2 l_2\cos\theta_2 - m_2 g l_2\cos\theta_2 \end{aligned} \tag{3}$$

Assuming there is no vicious force occurring in the system, the Lagrange equations can be written as:

$$u = (m_1 + m_2 + J_1\frac{1}{l_1^2})\ddot{Z}_1 + m_2\ddot{\theta}_2 l_2\cos\theta_2 - m_2 l_2\dot{\theta}_2^2\sin\theta_2 \tag{4}$$

and

$$0 = (m_2 l_2^2 + J_2)\ddot{\theta}_2 + m_2\ddot{Z}_1 l_2\cos\theta_2 - m_2 g l_2\sin\theta_2 \tag{5}$$

This can be rewritten as:

$$\ddot{\theta} = \alpha - \beta \tag{6}$$

where

$$\alpha = \frac{(m_2 l_2 g\sin\theta)(m_1 + m_2 + \frac{J_1}{l_1^2})}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2\cos\theta)^2}$$

$$\beta = \frac{(m_2 l_2\cos\theta)u + (m_2^2 l_2^2\cos\theta\sin\theta)\dot{\theta}^2}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2\cos\theta)^2}$$

and

$$\ddot{z} = \chi - \delta \tag{7}$$

where

$$\chi = \frac{(m_2 l_2^2 + J_2)u + ((m_2 l_2^2 + J_2)((m_2 l_2 \sin\theta)\dot{\theta}^2)}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2 \cos\theta)^2}$$

$$\delta = \frac{(m_2^2 l_2^2 g \sin\theta\cos\theta)}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2 \cos\theta)^2}$$

Since the objective is to keep the pendulum upright, it seems reasonable to assume that θ(t) and $\dot{\theta}$(t) will remain close to zero. The non-linear model is thus linearized, in order to simplify the analysis and design of the controllers, using these approximations: $\sin\theta \approx \theta$, $\cos\theta \approx 1$, $\theta x\theta \approx 0$, $\dot{\theta} x\dot{\theta} \approx 0$. Eq. (1), and Eq. (2) are reduced to:

$$\ddot{\theta} = \widetilde{\alpha} - \widetilde{\beta} \tag{8}$$

where

$$\widetilde{\alpha} = \frac{(m_1 + m_2 + \frac{J_1}{l_1^2})(m_2 l_2 g)\theta}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$\widetilde{\beta} = \frac{(m_2 l_2)u}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$\ddot{z} = \widetilde{\chi} - \widetilde{\delta} \tag{9}$$

where

$$\widetilde{\chi} = \frac{(m_2 l_2^2 + J_2)u}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$\widetilde{\delta} = \frac{(m_2^2 l_2^2 g)\theta}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

Choosing the states $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = z$ and $x_4 = \dot{z}$, we obtain the following state model

$$\dot{x} = Ax + bu \tag{10}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ K_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 \end{bmatrix}; b = \begin{bmatrix} 0 \\ K_2 \\ 0 \\ K_4 \end{bmatrix}$$

and the parameterts are:

$$K_1 = \frac{(m_1 + m_2 + \frac{J_1}{l_1^2})(m_2 l_2 g)}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$K_2 = \frac{-(m_2 l_2)}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$K_3 = \frac{-(m_2^2 l_2^2 g)}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

$$K_4 = \frac{(m_2 l_2^2) + J_2}{((m_2 l_2^2 + J_2)(m_1 + m_2 + \frac{J_1}{l_1^2})) - (m_2 l_2)^2}$$

3. FUZZY CONTROLLER FOR STABILIZING

This section discusses the design of feedback system of the FLS and system, which is used to control the stability and mobility of the wheelchair. Figure 4 demonstrates the block diagram to control its pitch angle, and to stop the motion to maintain the stability of two-wheels. It is evident that the FLS is

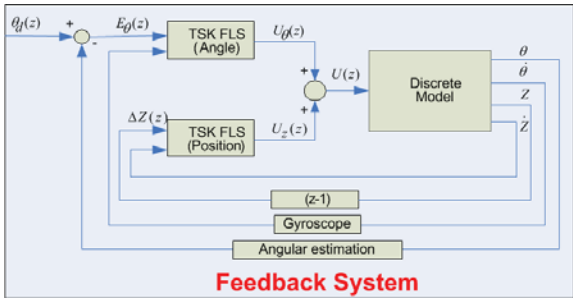

Fig.3:: Block diagram showing the pitch and wheel direction control based on the FLS scheme

designed to handle the discrete time model obtained by discretizing Eq. (10), with the sampling rate of 1 KHz. The inputs to the upper FLS are the angle error E_θ(z) and the angular velocity $\dot{\theta}$(z). Meanwhile, the force U_θ(z) is defined as the output. The objective of the upper FLS is to generate some force, so that the wheelchair can incline to its desired angle θ_d(z). In this case, θ_d=0. For the lower FLS, the one-time step difference between the distance and the velocity is inputted to the FLS. The output U_Z(z) is generated to stop the motion after a steady state has reached. The sum of U_θ(z) and U_Z(z) is then applied to the wheelchair (rear wheels).

The following subsections will provide brief overviews of type-1 TSK FLS, and interval type-2 TSK FLS, as used in the block diagram in Figure 4. Later on, the same type of FLS is deployed in the upper and lower blocks, in order to compare the effectiveness of the type-1 with type-2 designs.

3.1 Zero-Order Type-1 TSK FLS

In this section, two zero-order type-1 TSK models are considered with a rule base of 25 rules, and with each having 2 antecedents. The rules are shown in Table 1. The first model, as highlighted, is designed to achieve the desired angle, which is zero degree in our work, while the second model is used to stop the motion of the wheelchair after its steady state has reached.

Table 2:: Type-1 Fuzzy rules for controlling the angle and displacement.

The antecedents shown in Figure 5 are type-1 fuzzy sets. Meanwhile, the consequents presented in Figure

E_0 \ $\dot{\theta}$ / $\dot{Z}$ / ΔZ	NB	NS	ZO	PS	PB
NB	PM / PBB	PB / PBB	PB / PB	PBB / PB	PBB / PM
NS	PS / PBB	PM / PB	PB / PB	PB / PM	PBB / PS
ZO	NM / PM	NS / PS	ZO / ZO	PS / NS	PM / NM
PS	NBB / NS	NB / NM	NB / NB	NM / NB	NS / NBB
PB	NBB / NM	NBB / NB	NB / NB	NB / NBB	NM / NBB

6 are designed as crisp numbers to reduce computational complexity. The output of type-1 TSK FLS is obtained as:

$$y_{TSK}(x) = \frac{\sum_{i=1}^{M} f^i(x) y^i(x)}{\sum_{i=1}^{M} f^i(x)} \tag{11}$$

where $y^i(x)$ is the output of the i^{th} rule. and $f^i(x)$ is the rule firing level defined as:

$$f^i(x) = min\left[\mu_{F_1^i}(x_1), \mu_{F_2^i}(x_2)\right] \tag{12}$$

The parameter $\mu_{F_k^l}(x_k)$ denotes the membership function of the antecedent k in rule l.

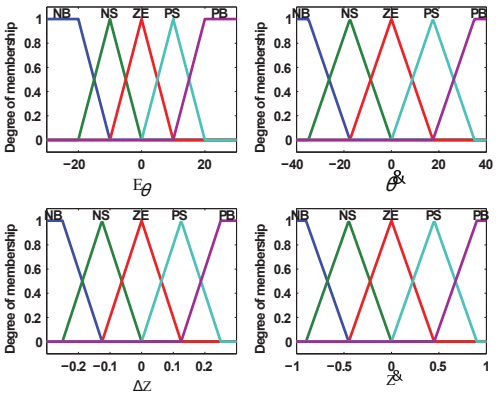

Fig.4:: Primary membership functions of the antecedents.

3.2 Interval Type-2 TSK FLS

A zero-order interval type-2 TSK model having the same rules and number of antecedents as the type-1 model. is considered. The difference between this model and Zero-Order Type-1 TSK is only the primary memberships of the inputs containing the uniformly shaded FOUs in the fuzzy sets, as depicted in Figure 7. Note that the secondary membership functions are interval sets.

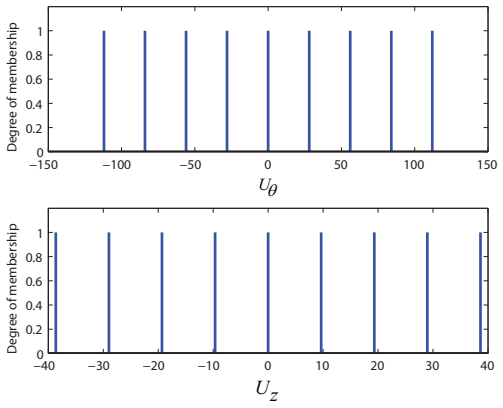

Fig.5:: The consequents.

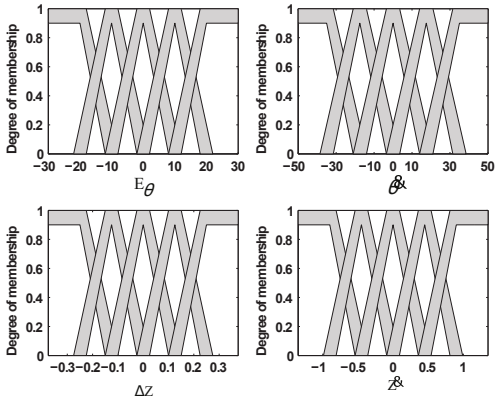

Fig.6:: Pictorial representation of a type-2 fuzzy set.

Let $\underline{\mu}_{\widetilde{F}_k^l}(x_k)$ and $\overline{\mu}_{\widetilde{F}_k^l}(x_k)$ denote the lower and upper membership functions for $\mu_{\widetilde{F}_k^l}(x_k)$ where $k = 1, 2$ (number of antecedents) and $l = 1, 2, 3, ..., 25$ (number of rules). For an interval type-2 TSK FLS, the results of the input and antecedent operations is an interval type-1 set, wherein the firing interval, is a set of $\left[\underline{f}^l, \overline{f}^l\right]$ that is determined by using a definition of the minimum t-norm as follows:

$$\underline{f}^i(x) = min\left[\underline{\mu}_{\widetilde{F}_1^l}(x_1), \underline{\mu}_{\widetilde{F}_2^l}(x_2)\right]$$

$$\overline{f}^i(x) = min\left[\overline{\mu}_{\widetilde{F}_1^l}(x_1), \overline{\mu}_{\widetilde{F}_2^l}(x_2)\right] \tag{13}$$

The fired output consequent $\mu_{\widetilde{B}^l}(y)$ of rule R^l can be obtained from the parameters given in Table 1 using the fuzzy rules. For a type reduction, and interval set determined by its two endpoints, this can be expressed as follows:

$$y_l = \frac{\sum_{i=1}^{M} \underline{f}^i y_l^i}{\sum_{i=1}^{M} \underline{f}^i} \tag{14}$$

$$y_r = \frac{\sum_{i=1}^{M} \overline{f}^i y_r^i}{\sum_{i=1}^{M} \overline{f}^i}$$

where M is 25 in this design. We finally defuzzify the interval set using the average of:

$$y(x) = \frac{y_l + y_r}{2} \tag{15}$$

4. ITERATIVE LEARNING CONTROL

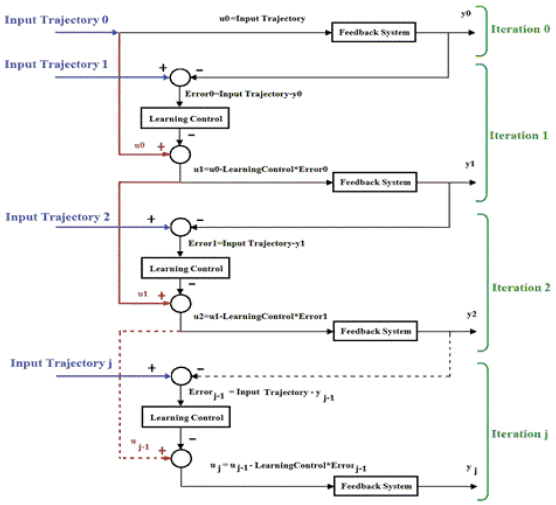

Fig.7:: Block diagram showing Iterative Learning Control scheme

4.1 Repetition domain formulation

The following summarises the ILC formulation originally developed in Phan and Longman [16] and later extended in Phan, Longman, and Moore [17]. Consider a system described as the general form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k), \end{aligned} \tag{16}$$

where A, B, C are the matrices describing the system dynamics. For simplicity, we consider the single-input single-output (SISO) case. Extension to the multiple-input multiple-output (MIMO) case is straightforward. The index k denotes the time t = kT , where T is an appropriate sampling interval. The vectors x(k) and y(k) are here denoted by the system state and output, respectively. The initial state x(0) and the process disturbance v(k) are assumed to be same from one repetition (or trial) to the next repetition. Define the following time history vectors:

$$\underline{y}_j = \begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(p) \end{bmatrix}; \underline{y}^* = \begin{bmatrix} y^*(1) \\ y^*(2) \\ \vdots \\ y^*(p) \end{bmatrix} \tag{17}$$

The $\underline{y}^*$ represents the desired output trajectory with p-time-steps long. In the above definitions, we assume that the time delay through the system is one-time step, i.e. the product CB is non-zero matrix. For any repetition j, the relationship between an input time history and the resulting output time history is given by

$$\underline{y}_j = P\underline{u}_j + \underline{w}, \tag{18}$$

where

$$P = \begin{bmatrix} CB & & & & \\ CAB & CB & & & \\ CA^2B & CAB & \ddots & & \\ \vdots & \ddots & \ddots & CB & \\ CA^{p-1}B & \dots & CA^2B & CAB & CB \end{bmatrix} \tag{19}$$

This equation packages the convolution sum solution of (16) for all p-time-steps in the input-output histories of a run. The vector w incorporates the effect of the unknown initial state x(0) and the unknown disturbance v(k). The tracking error for repetition j is

$$\underline{e}_j = \underline{y}^* + \underline{y}_j. \tag{20}$$

Define a backward difference operator $\delta_j(.)$ for any variable $\underline{z}$ as

$$\delta_j \underline{z} = \underline{z}_j - \underline{z}_{j-1}. \tag{21}$$

Thus, $\delta_j \underline{e} = -\delta_j \underline{y}$. Applying this operator to (18) yields

$$\underline{e} = \underline{e}_{j-1} - P\delta_j \underline{u}. \tag{22}$$

Notice that the unknown repeating initial condition and disturbance are eliminated by this operation. Iterative learning controllers designed from (22) will automatically compensate for any unknown repetitive disturbances regardless of any repeating initial condition. This equation forms the basis for the development of various ILC laws using modern state-space techniques.

4.2 ILC convergence conditions

A general (first-order) linear iterative learning control law has the form

$$\underline{u}_j = \underline{u}_{j-1} + L\underline{e}_{j-1}, \tag{23}$$

where L is a matrix of control gains chosen by the specific ILC design. Recognising that (23) can be written as $\delta_j \underline{u} = L\underline{e}_{j-1}$ the error propagation equation from one iteration to the next can be derived from (22) as

$$\underline{e}_j = (I - PL)\underline{e}_{j-1}, \tag{24}$$

where I is the identity matrix. One concludes from this that the tracking error at every time step of the p-time-step desired trajectory will go to zero as j tends to infinity, for all initial error histories, if and only if

all eigenvalues of the matrix $(I - PL)$ are less than one in magnitude,

$$max_i|\lambda_i(I - PL) < 1. \quad (25)$$

Some ILC laws have poor transients during the convergence process even if (25) is satisfied. One can ensure that the Euclidean norm of the error decreases monotonically for every iteration if L satisfies the sufficient stability condition

$$max_i\sigma_i(I - PL) < 1. \quad (26)$$

where σ_i is the ith singular value of $(I - PL)$, Longman [5]. Although (25) is the true stability condition, (26) is more practical.

5. ADAPTIVE ITERATIVE LEARNING CONTROL

In system development on automatic gain control of iterative learning control for repetitive behavior for better stability, In this research, we have developed an algorithm that uses Fuzzy logic to optimize the gain control of iterative learning control to get the best set point of the system. Fig. 9 shows automatic

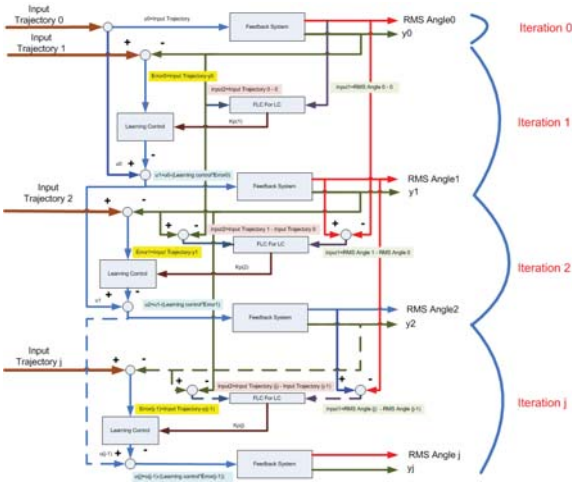

Fig.8:: Block diagram showing automatic gain control of iterative learning control scheme

gain control of iterative learning control scheme by using fuzzy logic control for adjusting gain of iterative learning control. In design fuzzy logic use 2 input, first input from rms error and second is iteration.

$$Input1 = Rmserrorofangle(j) - Rmserrorofangle(j-1), \quad (27)$$

$$Input2 = Iterationofangle(j) - Iterationofangle(j-1), \quad (28)$$

Two zero-order type-1 TSK models are considered in this section, with a rule base of 12 rules. The rules are shown in Table 3 and fig 10 ,11 and 12 shows membership function of a fuzzy set.

Table 3:: Type-1 Fuzzy rules for controlling the gain of Iterative Learning Control

Input1 / Input2	ZO	PS	PM	PB
ZO	ZO	PS	PM	PB
PS	PS	PM	PB	PVS
PB	PM	PB	PVS	PVB

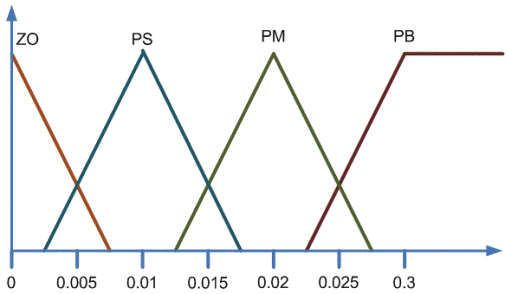

Fig.9:: RMS Angle of membership

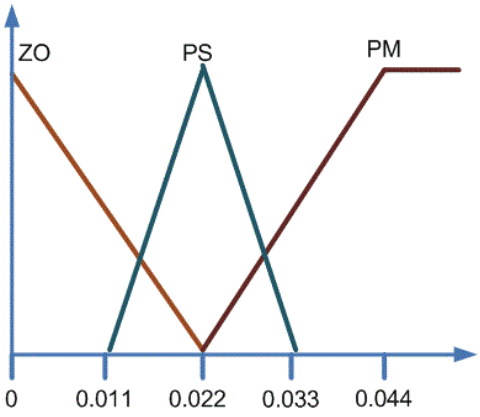

Fig.10:: Input Trajectory of membership

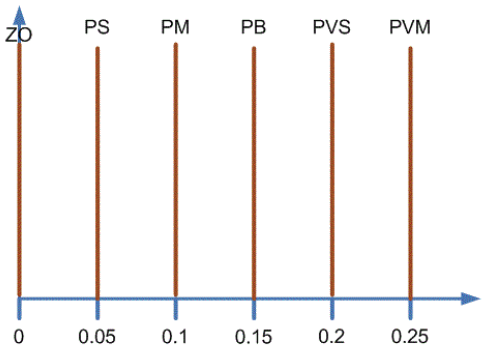

Fig.11:: The consequents for gain of Iterative Learning Control

6. SIMULATION RESULTS

Under the assumption that the initial angle is at -10 deg, the comparison of the performances of the two proposed controllers are shown in Fig. 13 and 14. The plot in the upper left corner in these figures shows how good they can track the desired angle.

It is obviously seen that same four lines of them can track the desired angle within a few seconds.

However, adaptive iterative learning control system has a better balancing and slightly better setting time than T1TSK and T2TSK.

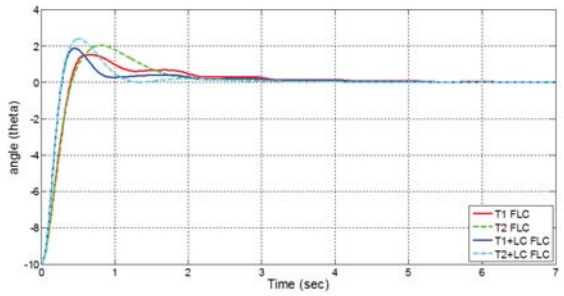

Fig.12:: Comparison of Type-1, Type-2 FLS and FLS Developed by learning control

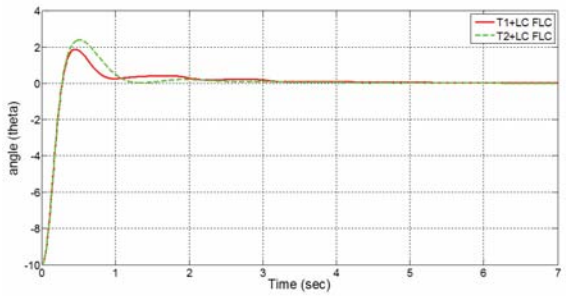

Fig.13:: Comparison of Type-1 and Type-2 FLS Developed by learning contro

In Fig. 15 and 16, RMS Error of angle and RMS Error of position in each learning cycle are shown. In more learning 300 cycles, the system has been stable. RMS error of angle of T1FLC has lower angle on 1.35 degree from 1.54 degree. It is lower 12.34% and T2FLC has lower angle on 1.39 degree from 1.57 degree. It is lower 11.47% and RMS error of position of T1FLC has lower displacement on 0.68 m from 0.95 m. It is lower 28.42% and T2FLC has lower displacement on 0.38 m from 0.65 m. It is lower 58.46%.

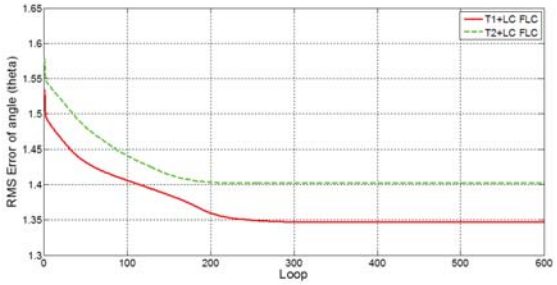

Fig.14:: RMS Error of angle in each learning cycle

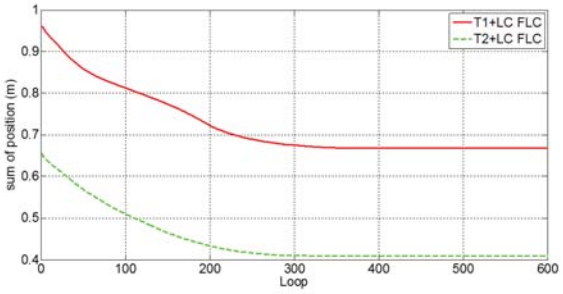

Fig.15:: RMS Error of position in each learning cycle

In Fig.17 the best set points of system for T1+LC FlC and T2+LC FlC are shown. In Fig.18, the best

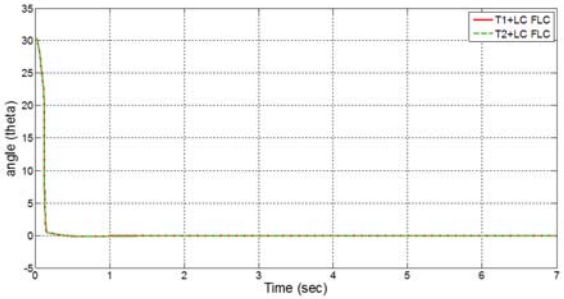

Fig.16:: Set point of system on T1+LC FLC and T2+LC FLC

learning cycles to use force for the system on T1+LC FlC and T2+LC FlC are shown. In Fig. 19, the angular velocities of the wheelchair are demonstrated in the upper right and lower right corners, respectively. In Fig. 20, the plot in the lower left corner gives us more information about the displacement of the wheelchair. Type-2 stops moving around 0.38 m from the original position. On the other hand, type-1 stays steady at 0.68 m from the original position. As

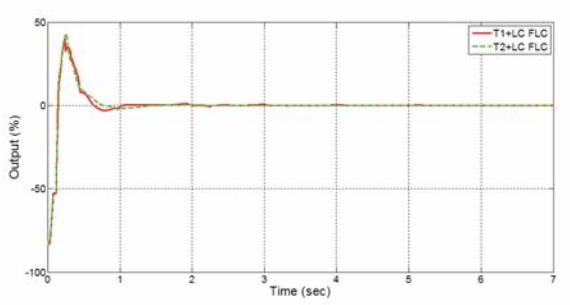

Fig.17:: Force applied to the wheelchair corresponding

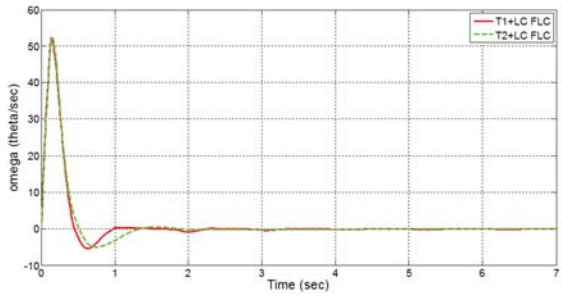

Fig.18:: Omega applied to the wheelchair corresponding

the result, type-2 performs better in the sense that it can reach the steady state and stop moving faster.

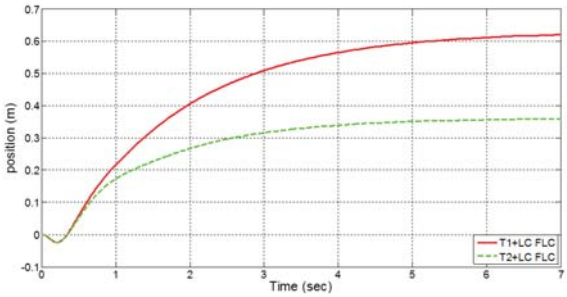

Fig.19:: Displacement applied to the wheelchair corresponding

7. CONCLUSION

In this paper, we have designed adaptive iterative learning control system by adjusting gain of the system correctly with the fuzzy logic control,The best point value of the system in 1 cycle to learning for feedback system, by the effect of control was determined by taking into accounts the issues:

1. to improve the performance in response to Iterative Learning Control.
2. to consider the trend of the test results between the first create original controller and new developed controller.

The performances using both controllers designs were demonstrated and compared. It was found from our simmution that Type-2+LC FLC performs more effectively, and provides better results with respect to stabilizing the wheelchair in an upright position and can reach the steady state faster but required higher exerted force to balance the wheelchair. Consideration for the hardware limitation remains as future work when the implementation of the design in the real world is needed.

8. ACKNOWLEDGEMENT

This work was supported by National Research Council of Thailand (NRCT).

References

[1] D. L. Kamen, R. R. Ambrogi, R. K. Heinzman, C. C. Langenfeld, M. A. Nisbet, S. B. Smith (III), and T. A. Brindley, "Medical improvements to a personal vehicle.," *U.S. Patent 6*, June 2002.

[2] D. L. Kamen, R. R. Ambrogi, J. D. Heinzmann, R. K. Heinzmann, D. Herr, and J. B. Morrell, "Control inputs for a balancing personal vehicle.," *J. Physiol*, vol. 369, p. 3, July 1985.

[3] B. Panomruttanarug and K. Higuchi, "Self-Balancing iBOT-Like Wheelchair Based on Type-1 and Interval Type-2 Fuzzy Control.," *11th International Conference on Electrical Engineering Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).*

[4] P. Chotikunnan and B. Panomruttanarug, "The Application of Fuzzy Logic Control to Balance a Wheelchair.," *Journal of Control Engineering and Applied Informatics*, 2015.

[5] R. W. Longman, "Iterative Learning Control and Repetitive Control for Engineering Practice.," *International Journal of Control*, vol. 73,p. 930-954, 2000.

[6] M. Q. Phan, R. W. Longman, B. Panomruttanarug and S. C. Lee, "Robustification of Iterative Learning Control and Repetitive Control by Averaging.," *International Journal of Control*, vol. 86,p. 855-868,2013.

[7] D. H. Owens, "Stability of linear multipass processes.," *Proceedings of the IEEE*,1977.

[8] M. Uchiyama, "Formulation of high-speed motion pattern of a mechanical arm by trial.," *Transactions of the Society for Instrumentation and Control Engineers*, 1978.

[9] E. Rogers, "Feedback and stability theory for linear multipass processes (Research Report).," *The Queen's University of Belfast, Department of Electrical and Electronic Engineering*,1987.

[10] K. L. Moore, "Iterative learning control for deterministic systems: Advances in Industrial Control.," *London: Springer-Verlag*,1993.

[11] Z. Bien and J.-X. Xu, "Iterative learning control: Analysis, design, integration, and applications.," *Boston: Kluwer Academic Publishers*,1998.

[12] A. Frueh and M. Q. Phan, "Linear quadratic optimal learning control (LQL).," *J. International Journal of Control*, vol. 73(10),p. 832–839,2000.

[13] M. Q. Phan, R. W. Longman and K. L. Moore, "A unified formulation of linear iterative learning control.," *Advances in the Astronautical Sciences*, vol. 105,p. 93–112,2000.

[14] H. Elci, R. W. Longman, M. Q. Phan, J.-N. Juang and R. Ugoletti, "Simple learning control made practical by zero-phase filtering: Applications to robotics.," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49(6),p. 753–767,2002.

[15] [15] B. G. Dijkstra and O. H. Bosgra, "Extrapolation of optimal lifted system ILC solution with application to a wafer stage.," *In Proceedings of the American Control Conference (2595–2600)*,2002.

[16] M. Q. Phan and R. W. Longman, "A mathematical theory of learning control for linear discrete multivariable systems.," *In Proceedings of the AIAA/AAS Astrodynamics Conference*, p. 740–746, Minneapolis, Minnesota,1988.

[17] M. Q. Phan, R. W. Longman and K. L. Moore, "A unified formulation of linear iterative learning control.," *Advances in the Astronautical Sciences*, vol. 105, p. 93–112,2000.

APPENDIX

L	The total Lagrangain
K_T	The total kinetic energy in the system
P_T	The total potential energy
g	Gravitational force
u	Input of the system
Z	Linear displacement of robot
$\dot{Z}$	Linear velocity of robot
$\ddot{z}$	Linear acceleration of robot
θ	Rotation angle of robot
$\dot{\theta}$	Angular velocity of robot
$\ddot{\theta}$	Angular acceleration of robot
Z_1	Linear displacement of wheel (Link1)
$\dot{Z}_1$	Linear velocity of wheel (Link1)
θ_2	Rotation angle of Link2
$\dot{\theta}_2$	Angular velocity of Link2
m_1	Mass of wheel (Link1)
m_2	Mass of Link2
J_1	Moment of inertia of wheel (Link1)
J_2	Moment of inertia of Link2
r_1	Radius of wheel (Link1)
l_2	The radius is the length of the rod (Link2)