Optimal Fuzzy Controller, using a Genetic Algorithm for a Ball on Wheel System

Péter Menich, József Kopják

Kandó Kálmán Faculty of Electrical Engineering, Óbuda University,
Bécsi út 96/b, H-1034 Budapest, Hungary
email: peter.menich@pentasoft.hu, kopjak.jozsef@kvk.uni-obuda.hu

Abstract: The process of building a fuzzy controller for the Ball on Wheel problem raised an issue with the optimization of the controller parameters. For non-linear systems the parameter values of the controller are based on a subjective estimate and a trial-and-error approach. This is a time-consuming human task, which could be automated. The information obtained from the visualized simulation results gave us the opportunity to increase the controllable area, using a genetic algorithm (GA), by tuning the parameters of the fuzzy membership methods.

Keywords: unstable; fuzzy; genetic algorithm; microcontroller; BLDC; simulink

1 Introduction

The control of an unstable non-linear system is always a challenging task. [1] Fuzzy controllers are well suited for such controlling applications [22] e.g., inverted pendulum [14] control of an unstable bioreactor [23]. Combining the fuzzy controller with various techniques opens up the possibility to increase the performance of the controller without significantly increasing the required computational power. A list of representative examples showing the huge scope for further development of controller design: predictive control [24], data-driven hybrid controller design [25], evolving fuzzy models [26], using lookup tables [29], cascade system structure [30].

Performance can also be improved by optimizing the controller. This can be achieved in several ways, of which the following are some examples: Multitasking genetic algorithm [27], using a Slime mold algorithm [28]

Our goal was to build a very simple fuzzy controller for educational purposes for the “ball on the wheel" problem, where the task is to keep a free-rolling ball on the top of a rotating wheel. Since our goal was to make the controller as simple as possible, we discarded complex solutions and focused on optimizing the controller parameters, i.e., the points of the triangular membership functions.

To measure the quality, we constructed a parameter space that assigned an ITAE (Integral of the time-weighted absolute error) [2] value to the initial speed and position of the ball. It shows the size and shape of the controlled area. We used a genetic algorithm to expand this controlled area as much as possible. Unfortunately, the full parameter space cannot be mapped to evaluate the fitness function, so it was reduced to five points. In these five points, the ITAE values were determined and the sum of these values gave the result of the fitness function. Our approach allowed us to cover a relatively large parameter space in such a way that it can be used to evaluate the fitness function of the genetic algorithm. The visualization allowed us to investigate how the selection of the five points influences the shape of the controllable area and thus the behavior of the controller.

2 Ball on the Wheel

The ball on the wheel problem is the following: a free-rolling ball must be kept on the upper unstable equilibrium of a wheel by controlling its rotation. Ignoring the dissipation, there are an infinite number of solutions when the ball is rolling with constant speed at the upper equilibrium. Calculating with dissipation the ball can be kept also in a constant rolling state where the g acceleration keeps the balance with the dissipation. The controller is planned to keep the ball standing at the upper equilibrium. In order for the controller to be able to do this, it must be able to handle the various disturbances, whether they are the result of an external force or the fact that the ball is not in a stationary position when it is lifted to the upper equilibrium.

It is a good demonstration platform for non-linear, unstable system control [3] [4]. In [5] [6] PID controller, in [7] linear controller with trajectory planning was used to demonstrate a solution for this problem.

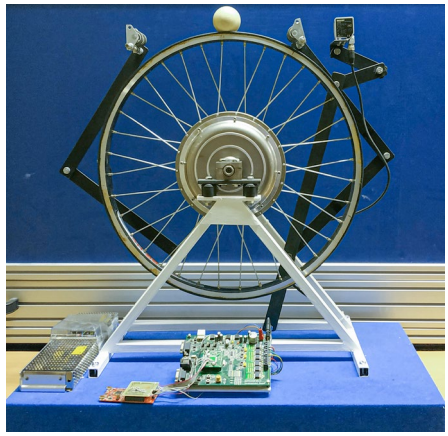

Figure 1
Ball on the wheel system

2.1 The Hardware

The hardware is based on an electric scooter wheel containing a BLDC motor with embedded HALL sensors. The driver is the MCLV development board from Microchip, the controller is an X2C development board using dsPIC33EP256MC502 microcontroller. The software uses a six-step control with PWM. The commutation is based on the inputs of the HALL sensors. The ball's position sensor is a BALLUFF BOD21M-LA04-S92 laser distance sensor. It has a 2.5 ms sample time with 4 samples moving average filter and an analog output.

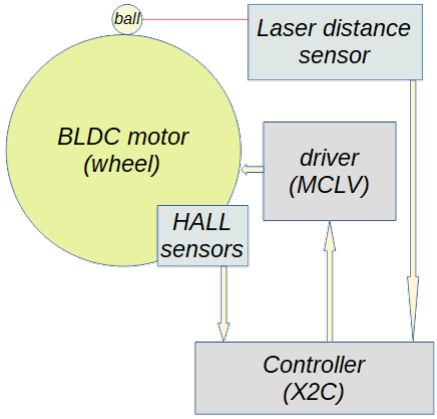

Figure 3
Schematic diagram of the hardware

2.2 The Model

We are using a fuzzy logic controller to balance the ball on the wheel, so the mathematical model building is not necessary, although it gives us the opportunity to examine the system in a virtual environment. From an educational perspective, it provides an opportunity for students to test other methods and controllers before they try them in a real environment.

2.1.1 Mathematical Model

The Lagrange formula: $L = T - V$ [9] (1)

Kinetic energy:

$$T = \frac{1}{2}\Theta_W \dot{\phi}_W^2 + \frac{1}{2}\Theta_B \left(\frac{r_W}{r_B}\left(\dot{\phi}_{pos} - \dot{\phi}_W\right)\right)^2 + \frac{1}{2} m_B \left((r_w + r_B)\dot{\phi}_{pos}\right)^2 \quad (2)$$

where φ_W is the current angle of the wheel and φ_{pos} is the ball position angle. The other parameters can be found in Table 1.

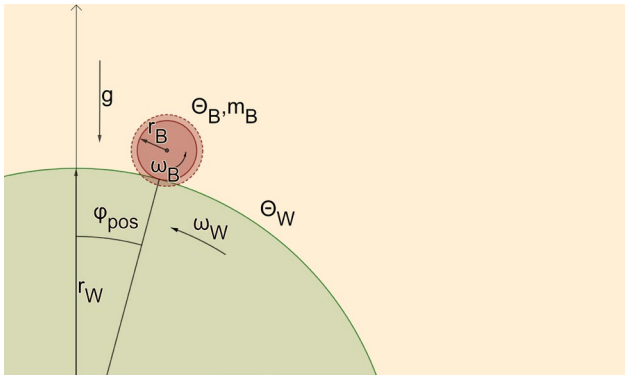

Figure 6
Schematic drawing of the Ball on the wheel

Potential energy (zero level is the center of the wheel)

$$V = -m_B g(r_B + r_w)cos(\phi_{pos}) \tag{3}$$

Generalized forces:

$$Q = \begin{bmatrix} Q_M - Q_W + Q_B \\ -Q_B \end{bmatrix} \tag{4}$$

Where,

Motor: $Q_M = \frac{K_M(U_a - K_U\dot{\phi}_W)}{R_a}$ (5)

where U_a is the armature voltage.

Wheel: $Q_w = K_{W1} \cdot sign(\dot{\phi}_W) + K_{W2} \cdot \dot{\phi}_W$ (6)

Ball: $Q_B = K_{B1} \cdot sign(\dot{\phi}_{pos} - \dot{\phi}_W) + K_{B2} \cdot \dot{\phi}_W$ (7)

The Lagrangian dynamics:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q \tag{8}$$

extracting (8) using (1) – (7) gives the following result:

$$\begin{bmatrix} \Theta_W + \Theta_B \frac{r_W^2}{r_B^2} & -\Theta_B \frac{r_W^2}{r_B^2} \\ -\Theta_B \frac{r_W^2}{r_B^2} & \Theta_B \frac{r_W^2}{r_B^2} + m_B(r_w + r_B)^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi}_W \\ \ddot{\phi}_{pos} \end{bmatrix} - \\ - \begin{bmatrix} 0 \\ m_B g(r_B + r_w)sin(\phi_{pos}) \end{bmatrix} = \begin{bmatrix} Q_M - Q_W + Q_B \\ -Q_B \end{bmatrix} \tag{9}$$

The mathematical model contains the following parameters of the physical system:

Table 1
Exact values of the parameters

Θ_W	0.30618	kg · m^2	moment of inertia (wheel)
Θ_B	0.0000731	kg · m^2	moment of inertia (ball)
m_B	0.203	kg	weight (ball)
r_W	0.27	m	radius (wheel)
r_B	0.0254	m	radius (ball)
R_a	0.3	Ω	armature resistance
K_M	2.093	V·s/rad	motor mechanic constant
K_{W1}	0.0023	N·m	Coulomb friction constant (wheel)
K_{W2}	0.0003	N·m·s	viscous friction constant (wheel)
K_U	2.518	N·m/A	motor electric constant
K_{B1}	0.045	N·m	Coulomb friction constant (ball)
K_{B2}	0.005	N·m·s	viscous friction constant (ball)

Using the values from Table 1, in (9) it gives the following differential equations formula:

$$\begin{bmatrix} \ddot{\phi}_{pos} \\ \ddot{\phi}_W \end{bmatrix} = \begin{bmatrix} 22.84 & -17.94 & -0.17 & -0.00235 & 7.115 \\ 0.6 & -56.34 & 0.01 & -0.00738 & 22.375 \end{bmatrix} \begin{bmatrix} sin(\phi_{pos}) \\ \dot{\phi}_W \\ sign(\dot{\phi}_{pos} - \dot{\phi}_W) \\ sign(\dot{\phi}_W) \\ U_a \end{bmatrix}$$

$$A = \begin{bmatrix} 22.84 & -17.94 & -0.17 & -0.00235 & 7.115 \\ 0.6 & -56.34 & 0.01 & -0.00738 & 22.375 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\phi}_{pos} \\ \ddot{\phi}_W \end{bmatrix} = A \cdot \begin{bmatrix} sin(\phi_{pos}) \\ \dot{\phi}_W \\ sign(\dot{\phi}_{pos} - \dot{\phi}_W) \\ sign(\dot{\phi}_W) \\ U_a \end{bmatrix} \tag{11}$$

2.2.2 Modelling in Matlab/Simulink

The model can be implemented in Simulink from the linear system of differential equations using the Kelvin-Thomson theorem [8]. The highest derivatives are separated on the left side, so the integrators can be used to get the lower derivatives.

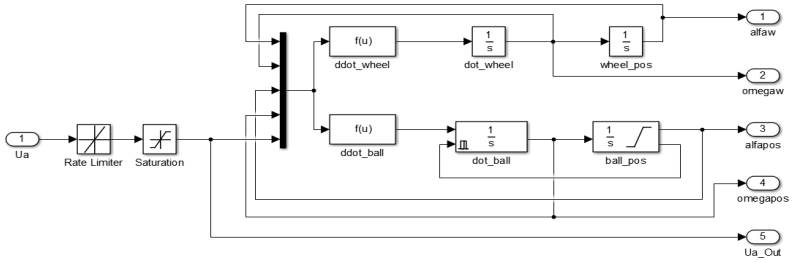

Figure 7

Simulink nonlinear model of the ball on the wheel problem

It is implemented in a subsystem. The input is the armature voltage and the outputs are all of the accessible parameters: the wheel's and the ball's position and angular velocity and the limited armature voltage. The input voltage is saturated to the nominal voltage of the hardware and the change rate is limited because the mathematical model enables fast oscillation of the input voltage, but it is not recommended to get it on the physical hardware. The ball's position is limited by bumpers, which is implemented by saturation. When the ball reaches the end position the velocity of the ball must be set to zero, so the integrator must be reset.

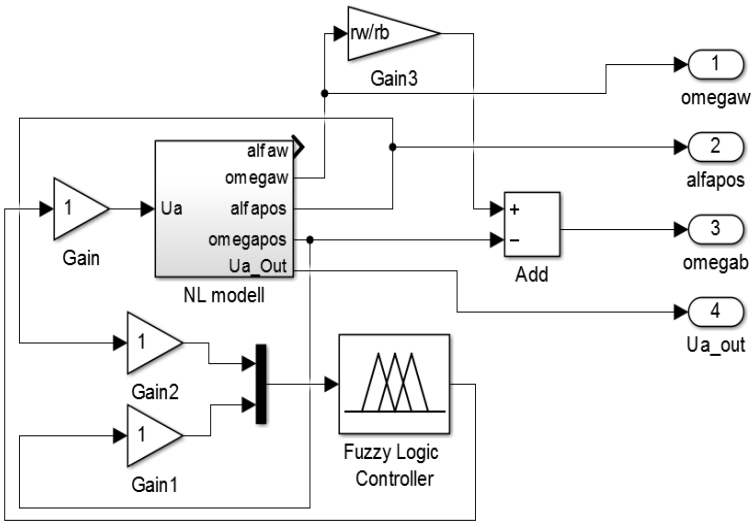

Figure 8

The ball on the wheel fuzzy controller

The current fuzzy controller configuration uses the ball position and velocity values for controlling the movement of the wheel set up the armature voltage.

Evaluating the quality of control

The different parameter setup gives very different behavior. To measure the quality of the control the ITAE (Integral of the time-weighted absolute error) function was implemented.

ITAE: $\int_0^\infty w(t) \cdot |e(t)|\, dt$ (12)

where w(t) is the time weighting and e(t) is the error function.

In our case the $w(t) = t$ (13)

and $e(t) = \alpha_{pos}(t)$ (14)

The ITAE (12) using (13) and (14) is $\int_0^\infty t \cdot \left|\alpha_{pos}(t)\right| dt$

The Simulink subsystem:

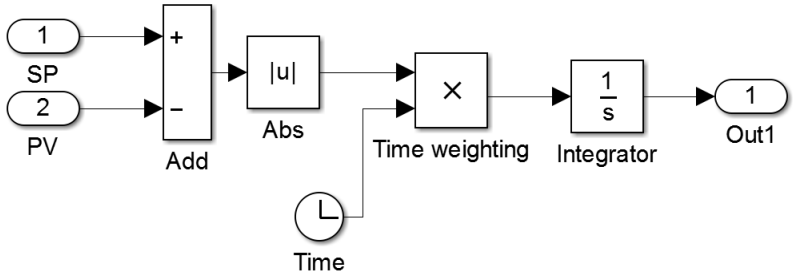

Figure 11
ITAE subsystem

The parameters are: SP – set point, PV – present value

2.2 Fuzzy Controller

The goal was to build a fuzzy controller as simple as possible, so the following restrictions were made:

- Using triangle membership functions
- Related to the symmetric problem, the fuzzyfication of the input values and the output defuzzyfication must be symmetric
- Using Mamdani method
- Using only two inputs
- Using Ruspini partitions [10]

Related to these restrictions the fuzzy controller can be defined by using 6 parameters. For each membership function (two inputs and one output) it needs two parameters. In Figure 7 the dotted lines mark the positions. The first parameter is the distance of line 'a' from center and the second is the distance line 'b' from the line 'a'.

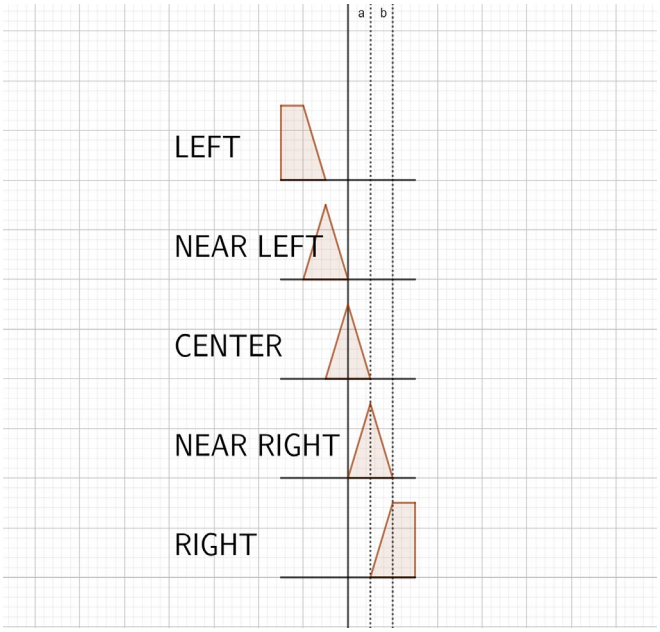

Figure 13
Fuzzy membership functions

The input 1 – ball position membership functions:

center: $\{[-a_{pos},0],[0,1],[a_{pos},0]\}$

near left, near right: $\{[-b_{pos},0],[-a_{pos},1],[0,0]\}$, $\{[0,0],[a_{pos},1],[b_{pos},0]\}$

far left, far right: $\{[min_{pos},1],[-b_{pos},1],[-a_{pos},0]\}$, $\{[a_{pos},0],[b_{pos},1],[max_{pos},1]\}$

The input 2 – ball angular velocity membership functions:

stop: $\{[-a_{vel},0],[0,1],[a_{vel},0]\}$

slow left, slow right: $\{[-b_{vel},0],[-a_{vel},1],[0,0]\}$, $\{[0,0],[a_{vel},1],[b_{vel},0]\}$

fast left, fast right: $\{[min_{vel},1],[-b_{vel},1],[-a_{vel},0]\}$, $\{[a_{vel},0],[b_{vel},1],[max_{vel},1]\}$

The output – armature voltage membership functions:

stop: $\{[-a_{out},0],[0,1],[a_{outl},0]\}$

low left, low right: $\{[-b_{out},0],[-a_{out},1],[0,0]\}$, $\{[0,0],[a_{out},1],[b_{out},0]\}$

high left, high right: $\{[min_{out},1],[-b_{out},1],[-a_{out},0]\}$, $\{[a_{out},0],[b_{out},1],[max_{out},1]\}$

The 6 parameters are: a_{pos}, b_{pos}, a_{vel}, b_{vel}, a_{out}, b_{out}

These parameters are used for both input membership functions (ball position, angular velocity of the position) and for the output membership function. [11-17]

3 Optimization

The model described in the previous chapter can be used to study the behavior of the system in a virtual environment with the given fuzzy controller parameters. We can study how the system behaves under different initial conditions. With given fuzzy parameters, we can plot the three-dimensional ITAE function image to study the controller behavior around the equilibrium.

Our goal is to design a controller that is less sensitive to disturbances, and to do this we want to tune the parameters of the fuzzy controller. The model also gives us the possibility to use it in the calculation of the fitness function of the genetic algorithm. To do this, we found that evaluating parameter space only at 5 points has a suitable effect on the shape of the controllable region. The sum of the ITAE function evaluated at the five points is used to calculate the value of the fitness function.

3.1 Genetic Algorithm

Genetic algorithms were inspired by natural evolution. The DNAs of the individuals are the current parameter values. The probability of survival and the chances of reproduction are based on the fitness function which determines the quality of the individual. [18-20]

DNA of each individual: $x = \begin{bmatrix} a_{pos} \\ b_{pos} \\ a_{vel} \\ b_{vel} \\ a_{out} \\ b_{out} \end{bmatrix}$

The genetic representation is the six parameters described above, and the fitness function is derived from the ITAE function.

The genetic algorithm feature of Matlab can handle restrictions in the following format: $A \cdot x \leq b$, where A and b can be defined, and x is the DNA's vector representation.

To have better performance the parameter space should be decreased as much as possible. The following restriction was made for the x parameter vector:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \cdot x \leq \begin{bmatrix} 0.5 \\ 15 \\ 24 \\ -0.01 \\ -0.01 \\ -0.01 \\ -0.01 \\ -1 \\ -1 \end{bmatrix}$$

This means that position must be in [-0.5,0.5] the angular velocity in [-15,15] range and the output must be in [-24,24] interval. The minimum distance between points must be 0,01 for input and 1 for output.

3.2 Fitness Function

The previously defined ITAE function measures the quality of the controller's response for the initial state, but our goal is to define the controller parameters to have a reasonable control near the area of the equilibrium. The following three parameters can describe the actual state of the system:

- Position of the ball
- Angular velocity of the wheel
- Angular velocity of the ball

Certainly, other linear combinations of these three parameters could be used also.

Since the goal is to have the ball standing at the equilibrium, only the first two parameters are used for the quality measurement. So, the first coordinate of the parameter space is the ball's position the second one is the speed of the wheel. The simulation starts with these parameters where the ball lateral speed is zero. In this case the initial angular velocity of the ball: $\omega_b = \frac{r_w \cdot \omega_w}{r_b}$

Unfortunately to evaluate the result of the fitness function for the whole parameter space is not possible, so only some predefined points were examined to check the size of the controllable area near the equilibrium. The redesigned simulation for the optimization has five parallel simulations and the fitness function is the summary of the ITEA values. The minimization criteria will cause the algorithm to hold the ball in the equilibrium at as many points out of five as it can. The best case is when all of them are controllable.

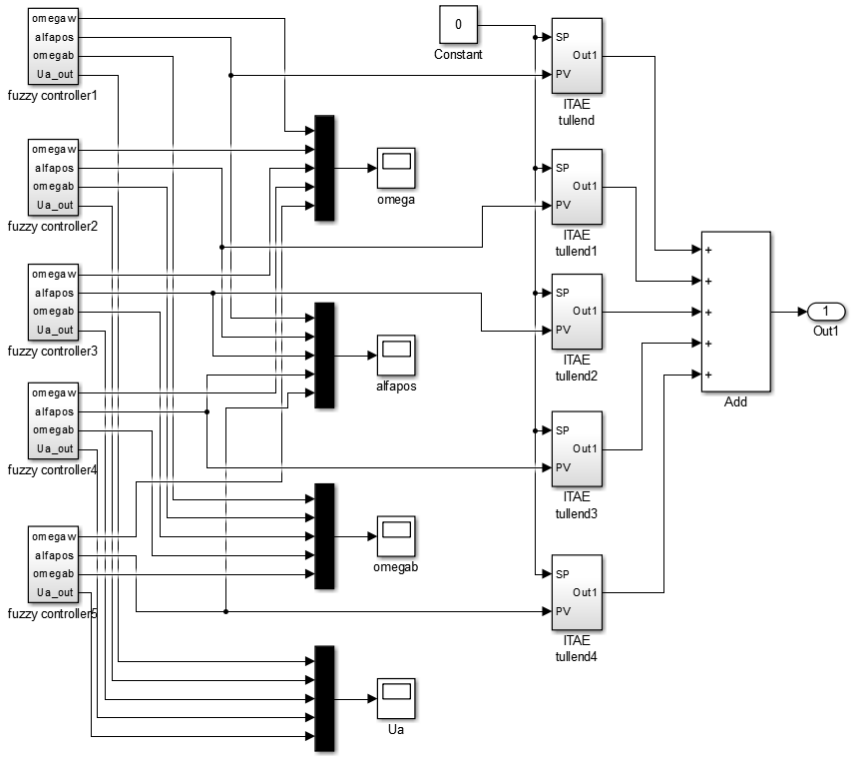

Figure 15
Five parallel simulation

3.3 Visualization

The visualization of the parameter space gives the opportunity to examine the behavior of the controller. To visualize we used the function $f: \mathbb{R}^2 \to \mathbb{R}_0^+$, where the domain is the initial position of the ball and the initial angular velocity of the wheel $(\alpha_{pos} \times \omega_w)$, the co-domain is the ITAE result.

Figure 9 shows the controllable area for a basic fuzzy membership function which was set similarly to Figure 7. The controllable area is where the value of the ITAE function is low (blue). It can be seen that the narrow path can be controlled mainly where the signed summary of the kinetic and potential energy of the system is near zero.

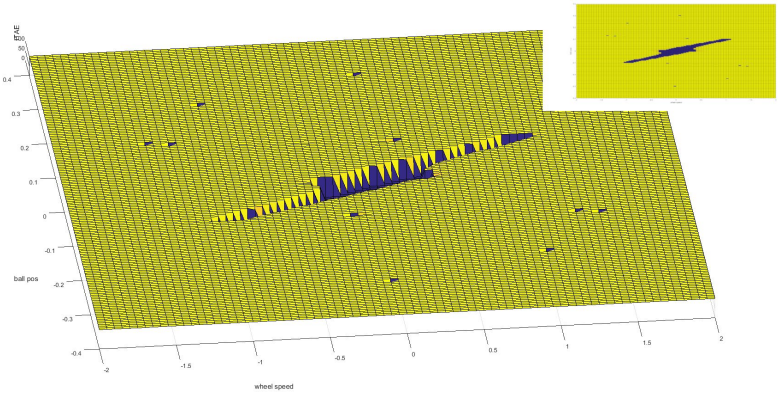

Figure 17
Controllable area visualization for simple equidistant fuzzy membership function

3.4 Expanding the Controllable Area

Unfortunately, the evaluation of the ITAE for the whole parameter space, is not possible. To get an acceptable runtime result we had to reduce the number of points to evaluate. The evaluation of five different points gives a big enough impact on the desired shape of the controllable area and provides acceptable calculation time. The summary of the ITAE values of these points gave the result of the fitness function. The implementation can be seen in Figure 8. Expanding the area on the ball position axis reduced the control on the other (Figure 10).

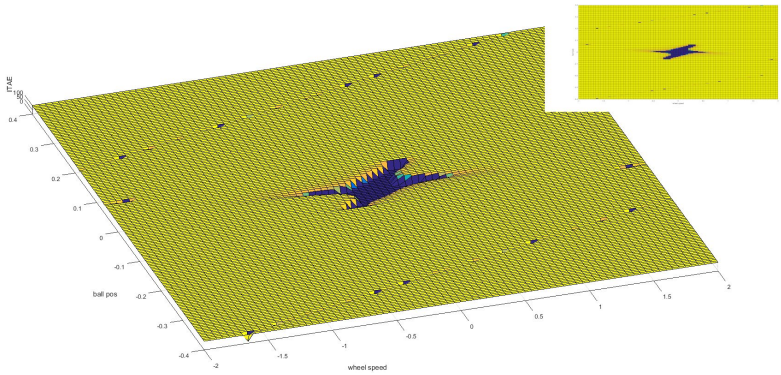

Figure 19
Expanded ball position area near to zero wheel speed

The result of the wheel speed insensitivity gives a narrow field for the ball's position (Figure 11).

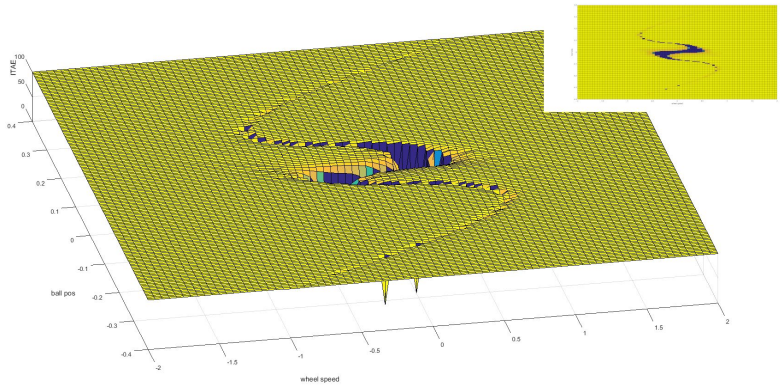

Figure 21
Narrow field

Putting the ball in a controllable position is not part of the controlling task. It can be done by a simple algorithm. Roll the ball on the bumper to get enough kinetic energy to "climb" up, turning the kinetic energy into potential energy. Near the equilibrium, the controller can be turned on. The path to the central area in Figure 11 gives us the possibility of putting the ball near to the equilibrium by the controller from the given state. We implemented this method on the physical system. Spinning the ball on the bumper using the right wheel speed for less than a second is enough for us to be able to turn on the controller and then it can put the ball to the right position and hold it there.

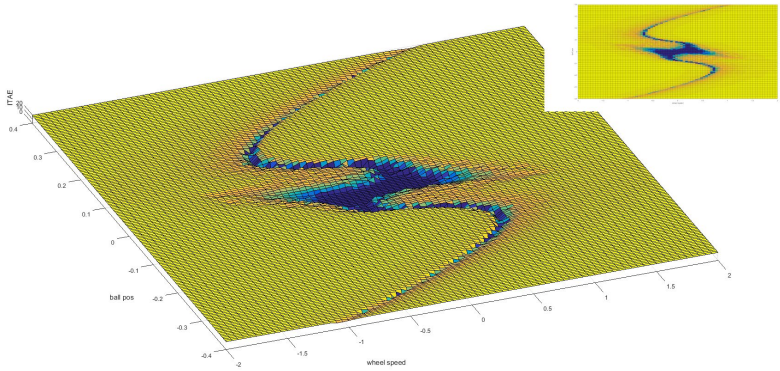

Figure 23
Big enough area with path to the center

4 Implementation

The original hardware was built by Artúr Kasovitz and Imre Dobany. The wheel's suspension was rigid so it had to be modified. The controller board was extended by the X2C module because it is our standard educational platform. [21]

For the software implementation, we used the C language on the MPLABX development environment by Microchip. With the given restrictions a very simple fuzzy controller was enough to handle the problem.

Since the primary goal of the project is to use the fuzzy controller for educational purposes, we have always tried to keep the controller as simple as possible, so we focused on the controller parameters. The mathematical model and the computer simulation allowed us to visualize the parameter space, i.e., how the controller behaves for a given initial ball position and ball speed. The visualization made it clear in which directions the controllable area needed to be increased. For this purpose, we constructed a simplified fitness function using only 5 points in the parameter space. Thus, using the mathematical model and the genetic algorithm to set up the parameters, we were able to maximize the performance of our controller and keep the implementation easy.

The dataflow of the software can be seen in Figure 13.

The software uses the laser distance sensor and the HAL sensors as input signals. The laser distance sensor is analog, so it is transformed to a digital value by the 12-bit A/D converter of the microcontroller. The position change gives the angular speed information. The output of the fuzzy controller is a signed armature voltage value. The sign is the direction information, and the absolute value gives the PWM duty cycle. The commutation event of the 6 steps controller is based on the signals of the HAL sensors and the direction information from the fuzzy controller. The output signal is controlled by the PWM duty cycle.

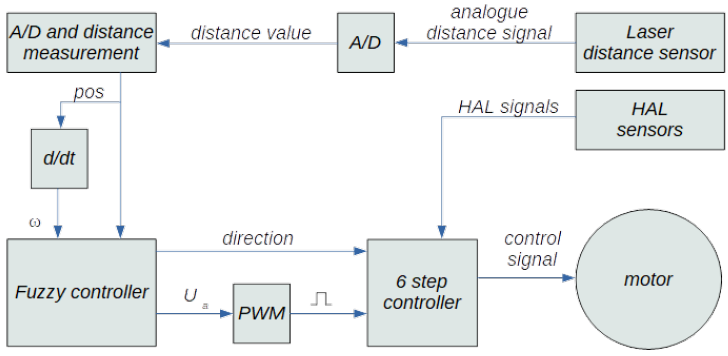

Figure 25
Software dataflow

Running the test can be seen on YouTube: https://youtu.be/hBOBb6hBWXw

Conclusions

Our goal was to build a relatively simple, fuzzy controller, for the implementation of the "ball on the wheel" system. To achieve this, we had to find a good parameter setup for the membership functions. We used a mathematical model and Matlab and Simulink tools to build up a virtual environment. In this system, we defined a relatively simple parameter space and fitness function, which was drawn as a $\mathbb{R}^2 \rightarrow \mathbb{R}$function. In this system we used a genetic algorithm to find a good setup for our controller parameters. An additional advantage of the visualization was that we found an alternative way for the initial lifting of the ball.

Acknowledgement

This work was supported by Óbuda University. We would like to thank to Artúr Kasovitz, Imre Dobány, Máté Gyimesi for their work on the hardware.

References

[1] Gunter Stein: Respect the unstable. Respect the unstable. IEEE Control Systems, (2003) 23(4), pp. 12-25

[2] Fernando G. Martins: Tuning PID controllers using the ITAE criterion, International Journal of Engineering Education, (2005) Vol. 21, No. 5, pp. 867-873

[3] H. K. Khalil: Nonlinear Systems. Upper Saddle River, (1996) NJ 07458: Prentice Hall, second ed.

[4] J. E. Slotine, W. Li: Applied Nonlinear Control, (1991) Englewood Cliffs, NJ: Prentice - Hall, Inc.

[5] M. T. Ho, J. M. Lu_ "H_∞ PID controller design for Lur'e systems and its application to a ball and wheel apparatus, International Journal of Control, (2005) Vol. 78, No. 1, pp. 53-64

[6] M. Ho, H. Lin: Balance Control of Ball and Wheel Systems via Feedback Linearization, (2006) Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, pp. 3926-3931

[7] S.-O. Lindert, C. Höfler, K. Schlacher: Nonlinear controlling using Raspberry PI (Nichtlineare Regelung mit einem Raspberry PI.) e & i Elektrotechnik Und Informationstechnik (2018) 135(3) pp. 286-293

[8] R. Herman: Solving differential equatins using Simulink, (2019) Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License

[9] J. H. Ginsberg: Advanced Engineering Dynamics second edition, (1998) Cambridge University Press

[10] Enrique H. Ruspini: A new approach to clustering, (1969) Information and Control Volume 15, Issue 1, pp. 22-32

[11] L. Kóczi T., D. Tikk: Fuzzy systems (Fuzzy rendszerek), (2000) Typotex Kiadó, Budapest

[12] M. Takács, Modified Distanece Based Operators in Fuzzy Approximate Reasoning, (2006) IEEE International Cenference on Mechatronics, Budapest pp. 297-299

[13] Wang, L.-X.: A Course in Fuzzy Systems and Control. (1997) Englewood Cliffs, NJ: Prentice-Hall

[14] Wang, L.-X.: Stable Adaptive Fuzzy Controllers with Application to Inverted Pendulum Tracking, (1996) IEEE Trans. on Syst., Man and Cyberneticspart B, Vol. 26, pp. 677-691

[15] Mathworks-MATLAB: Fuzzy Logic Toolbox, https://nl.mathworks.com/products/fuzzy-logic.html [Online] Available: https://nl.mathworks.com/products/fuzzy-logic.html

[16] B. Kosko: Fuzzy engineering, (1997) Prentice-Hall

[17] T. Takagi, M. Sugeno: Fuzzy Identification of Systems and its Applications to Modeling and Control, (1985) IEEE Trans. on Systems, Man, and Cybern., SMC-15(1) pp. 116-132

[18] A. Álmos, S. Győri, G. Horváth, A. Várkonyiné Kóczy: Genetic algorithms (Genetikus algoritmusok), (2002) Typotex Kiadó, Budapest

[19] William M. Spears: Evolutionary Algorithms, (2000), The Role of Mutation and Recombination. Springer, Berlin, Heidelberg, New York, (Natural Computing Series)

[20] Mathworks-MATLAB: ga Find minimum of function using genetic algorithm ,https://nl.mathworks.com/help/gads/ga.html [Online] Available: https://nl.mathworks.com/help/gads/ga.html

[21] X2C+ Development board, http://x2c.microstickplus.com/ [Online] Available: http://x2c.microstickplus.com/

[22] M. Olivares, P. Albertos: Fuzzy PD control of an unstable system, IFAC Proceedings Volumes Vol. 33, Issue 4, (2000) pp. 217-222

[23] M. Galluzzo, C. Cirino: Sliding Mode Fuzzy Logic Control of an Unstable Bioreactor (2013), Chemical engineering transactions Vol. 32, (2013) pp. 1213-1218

[24] A. Reda, A. Bouzid, J. Vásárhelyi: Model Predictive Control for Automated Vehicle Steering, Acta Polytechnica Hungarica Vol. 17, No. 7, (2020) pp. 163-182

[25] R.-C. Roman, R.-E. Precup, E. M. Petriu: Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems, European Journal of Control, Vol. 58, (2021) pp. 373-387

[26] R. Precup, T. Teban, A. Albu, A. Borlea, I. A. Zamfirache and E. M. Petriu: Evolving Fuzzy Models for Prosthetic Hand Myoelectric-Based Control, IEEE Transactions on Instrumentation and Measurement, Vol. 69, No. 7, (2020) pp. 4625-4636

[27] D.-R. Wu, X.-F. Tan: Multitasking genetic algorithm (MTGA) for fuzzy system optimization, IEEE Transactions on Fuzzy Systems, 28(6), (2020) pp. 1050-1061

[28] R.-E. Precup, R.-C. David, R.-C. Roman, A.-I. Szedlak-Stinean, E. M. Petriu: Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using Slime Mould Algorithm, International Journal of Systems Science (2021)

[29] A. Turnip, J. H. Panggabean: Hybrid Controller Design based Magneto-rheological Damper Lookup Table for Quarter Car Suspension, International Journal of Artificial Intelligence, vol. 18, no. 1, (2020) pp. 193-206

[30] T. Haidegger, L. Kovács, R.-E. Precup, B. Benyó, Z. Benyó, S. Preitl: Simulation and control for telerobots in space medicine, Acta Astron., Vol. 181, No. 1, (2012) pp. 390-402