Tele Operation of a Humanoid Robot using Fuzzy Control and Kalman Filter

A. Cela[(1)], L. M. Bergasa [(2)], R. Barea [(2)], E. López [(2)], A. Rodríguez [(2)]

***Abstract-* In the field of humanoid robotics, to implement a tele operate system to control a mini humanoid robot, as a first step for further development and study of algorithms for motion and human walking can be interesting. This paper gives the first guidelines on the implementation of a suit with 8 sensors, 6 in the lower extremities, which are resistive linear potentiometers and 2 more in the arms based on digital electronic accelerometers. Communication to the robot, controller through wireless transmission, is performed using two Zig-Bee RF configurable modules.**

The humanoid robot is controlled by a processor of medium capacity. This allows further development and simplification of software, such as controlling the number of wireless data, reducing the delay times of the algorithms, control code simplification, optimization of variables control, etc. The goal is to replicate the suit movements in the robot. Therefore, is necessary to control the stability of the robot so it does not fall if it executes the different actions. This is carried out via a feedback control system with an accelerometer placed on the robot's back. A Kalman filter algorithm reduces the signal noise and estimates the state. This information is sent to a two-input fuzzy controller (tilt angle and angular velocity) that control two servomotors to regulate the balance of the robot.

I. Introduction

For decades researches have studied different methods to reproduce their movements in electromechanical systems and even robotic systems that allow not only copy but improve and refine these movements for different uses for the welfare of humanity. The tele operation of construction machinery, mining machinery, chemical industry, provide protection and increase the maneuverability of the people, but robots in medicine require strict control to do surgery.

In medicine is very important the tuning of tele operator movements, algorithms that reduce noise and natural oscillations, are used for that. Also bio signals need be filtered [1] and Kalman filters are normally used for that purpose.

In addition, a reliable way to data transmission, wired or wireless, needs to be study as well as the information sent by the tele operator, avoiding unnecessary interruptions to the controller.

Finally, to choice an optimal controller for this application is very important. The Fuzzy controllers are need for multiple applications [2], [3], or to facilitate any previous controller [4].

That is a multidisciplinary controller, moreover, when the ultimate goal is to implement algorithms that allow to walk to a humanoid robot.

There are different techniques to capture human movements [5], as visualization using cameras [6], with sensors in the body (biomechanical) [5] or external [4], or computer animation using move descriptors [7]. These techniques reproduce basic motions (circle, lines, etc) and their parameters, and whole human movements are described with these basic motions [6], but it requires huge number of variables [5].

For this research we used some sensors in a mechanical suit that can be operated by a person. The associated movements active the sensors and the electronic system can read the data and send this information to the control system.

This paper presents a preliminary work and shows how to use a simple nonlinear control for balance a robot. However, it is important to remember that the optimization of nonlinear systems are computationally expensive, even more, if they need to solve a high degree of freedom system, such as a human robot [7], [4].

Furthermore, the sensors data are treated as joint variables rather than Cartesian variables [7]. Second option is more efficient, but, in this case is computationally impossible.

II. Tele operator suit

The tele operation suit has 6 sensors in the legs and 2 sensors on the arms, as it is shown in Fig. 1. The sensor legs are linear resistive. The sensor arms are accelerometers measuring in the x axis. For sensors of the arms bracelets have been used instead of joints.

Figure 2 shows the degrees of freedom (DoF) at each joint of the legs. The thigh points have two DoF, while each knee and arm joint has one. This mechanical system is not very efficient, but future implementations will improve its mechanical structure and electronics, and will increase the number of joints.

A. Resistive sensors and accelerometers

The 6 linear potentiometers are used to measure the angle of the lower suit joints. These reduce the computational cost because their response is linear, and the relationship between sensor voltages and the angle of inclination of the joint can be easily calculated.

(1) A. Cela is with the Ecuadorian Robotic Federation, National Polytechnic of Ecuador. andres.cela@alu.uah.es.

(2) L. M. Bergasa, R. Barea, E. López and A. Rodríguez are whit the Electronics Department Polytechnic Scholl, University of Alcala, Pta San Diego, s/n 28801, Alcalá de Henares, Madrid, Spain. bergasa, barea, elena@depeca.uah.es rodriguezfernndez@gmail.com

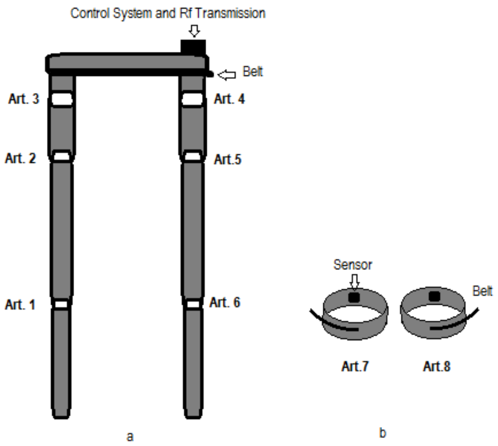

Figure 1. Tele operator suit. a) Legs suit. b) Arms suit

Figure 2 shows the joints in the legs suit and the associate motion for each one.

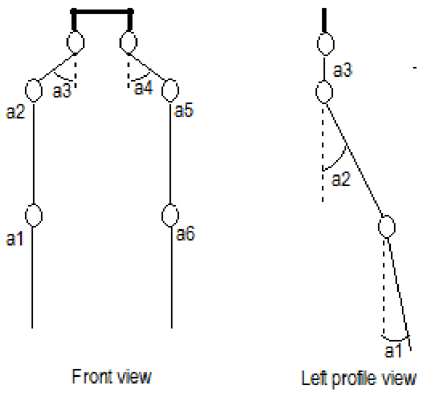

Figure 2. Degrees of freedom of the joints.

The typical output range for each sensor is from 0 to 1024, and it is obtained by an 10-bit analog channel using, 0 to 0 ° and 1024 to 180º. This data passes through a filter which scales the data from 0 to 180, that is the final data with which the servo motors of the robot are controlled, thus it considerably reduces noise and gives a resolution of 1º for moving motors.

The processor reads the 6 resistive sensors and 2 accelerometers and submits the information to the receiver. The data from the accelerometers arms follow the same procedure.

B. *Data Processing suit.*

It is necessary to send data from the sensors suit to the robot. After performing some tests, we considered sending RF (serial) protocol using Zig-Bee, because ensures the data frame. We also studied how to reduce this data frame.

When sending data in a natural way, a processor considers each data as a character, for example, if you wish to send the number "5", this leaves in the processor a 5 in ASCII, B00010101, instead of B00000101, which corresponds to 5 in binary, this causes that sent data frame grows.

In the worst case it is sent 3 digits for each sensor and a general heading, so the reader can recognize the start frame and a header for each sensor data and a finish indicator. Bellow we show an example data frame.

Xa170b180c180d180e180f180g180h180Y

This frame contains 48 bytes, whit a transmission rate of 57600bps, it results in approximately 8.42 ms. We improve this time reducing the data frame length. Then, the data is sent, not in ASCII format but in binary. This allows each byte of the frame is hosting the sensor data and sends only 8 bytes data and a header byte, (9 bytes); *ÿabcdefgh*, *where ÿ* is the header and each letter contains the sensor data A0 to A7. The binary value of ÿ = 255 never is taken. Whit this strategy the time required to send a data frame is 1.56ms, which reduces 5.3 times the delay of the first frame.

B. *Board Reading and transmission suit*

The board used for reading and sending the sensors data is a free hardware and software from Arduino-Fio. The sensors are read by the ADC module, it has 8 channels of 10 bits and these channels are those that have limited the number of used sensors. This board is depicted in Fig.3.

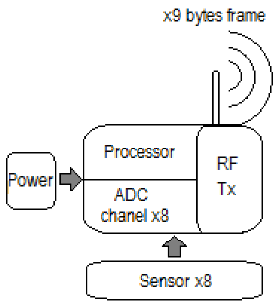

Figure 3. Processing system suit.

III. ROBOT CONTROL SYSTEM

The robot system consists of servo motors, balance sensor (accelerometer) and control board.

A. *Servomotor system*

A set of 16 actuators form the body of humanoid robot, but only 10 or 12 of them are controlled.

They are designed for this system, have metal gears and provide greater strength. They are controlled by a typical servomotor PWM control signal as we show in Fig.4.

Each servomotor starts in a known position and is internally controlled by PID control with resistive feedback. The servomotor gets the goal position and does not require feedback of the general system.

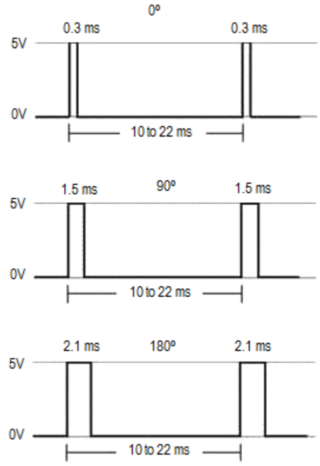

Figure 4. PWM control signal for a servomotor.

B. Balance feedback by Kalman filter

The balance feedback is done through a sensor located in the back of the robot. This is a 2g accelerometer, which generates an analog signal which is read by the processor. The state vector of the sensor contains orientation in 3-axes called Euler angles [8], but we only used one of them for control. Furthermore, the balance control system is Sugeno Fuzzy controller. Due to this is sensitive to noisy inputs, it is necessary to filter the sensor signal to improve the response of the controller.

The Kalman filter is a recursive algorithm that works well with systems corrupted by noise [9]. This filter is commonly used to directly estimate the orientation [8] or correct small misalignments in accelerometers, INSs, Inertial Motions Units (IMU) [10], [11] and the own sensors.

We can see in Fig. 5 the complete filter system, where A(k), B(k), C(k) are matrices, V (k) and W (k) are noise Stochastic Processes of system and measurement respectively.

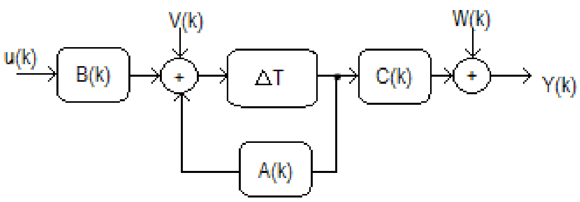

Figure 5. Kalman Filter.

The equations of this system are:

$$x_{k+1} = Ax_k + Bu_k + v_k$$
$$y_k = Cx_k + w_k \quad (1)$$

Where the following noise processes are applied:

$$Q_k = E\{v_k * v_k{}^T\}$$
$$R_k = E\{w_k * w_k{}^T\} \quad (2)$$

The solution is given in two parts, the first one is given by the time update (state estimation or prediction) and the second for a measurement update (correction, monitoring and updating) [9], [12].

The Q and R matrices are generally elected in an experimental way trial and error method [9], in this case they are:

$Q = \sigma v^2 = 0.03\,; \quad R = \sigma w^2 = 0.25$

The matrices are:

$B = 0; \quad C = 1; \quad A = 1;$

With initial conditions:

$x(o) = First\ sensor\ data$ and $P(o) = Q = 0.03.$

Next steps are applied to solve the filter.

State prediction:

$$x'_{k+1} = A\hat{x}_k + Bu_k$$
$$P'_{k+1} = AP_kA^T + Q \quad (3)$$

Gain K estimation:

$$K_{k+1} = CP'_{k+1} * [CP'_{k+1}C^T + R]^{-1} \quad (4)$$

The observation is to measure the new sensor data [10].

$y_{k+1} = new\ sensor\ measure$

The estimated state depends of new measurement data [10].

$$\hat{x}_{k+1} = x'_{k+1} + K_{k+1}[y_{k+1} - Cx'_{k+1}] \quad (5)$$

The difference $y_{k+1} - Cx'_{k+1}$ in (5) is called the measurement innovation and reflects the difference between the predicted and the current measurement [9], [11].

Update covariance P matrix will be given by:

$$P_{k+1} = (I - K_{k+1}C) * P'_{k+1} \quad (6)$$

Where:

$\hat{x}$ is the Estimated State.

x' is the Previous State.

P' is the Previous covariance P matrix.

Replacing the values in (3), (4), (5) and (6) we will have:

$x'_{k+1} = \hat{x}_k$

$P'_{k+1} = P_k + 0.03$

$K_{k+1} = P'_{k+1} * [P'_{k+1} + 0.25]^{-1}$

$y_{k+1} = to\ measure\ sensor$

$\hat{x}_{k+1} = x'_{k+1} + K_{k+1}[y_{k+1} - x'_{k+1}]$

$P_{k+1} = (1 - K_{k+1}C) * P'_{k+1}$

This algorithm applied to the sensor data produces the results of Fig. 6, where we can see two signals, the sensor data and the filtered data in green and black respectively.

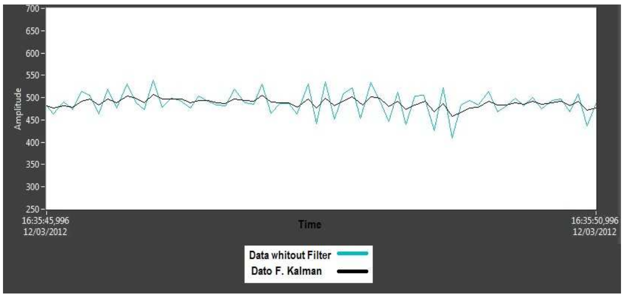

Figure 6. Kalman filter applied to the sensor data.

We have an acceptable response of the filter, whereas the noise of the data is not from the sensor but the movement of the robot's mechanical system.

C. Fuzzy Control of the robot balance

A fuzzy controller with Takagi Sugeno method (TS) has been implemented to control the balance of the robot. This controller does not use many processor resources and gives good results in real time. [13]

The TS method is sensitive to noise in the input noise, but filtering data using Kalman filter we dealt to reduce this problem. The basic idea is that the TS functions are good local control functions for the fuzzy region described by the corresponding rule premise [2].

The rules are built using the format suggested by [14], this rules defines the inputs, outputs, the implications format and the connected functions. However, some authors use Genetic Algorithms or other computational algorithms for tuning the rules and membership functions [15]. In our case these have been established in an experimental way.

$$R^i\text{: } \textbf{IF } f(x_1 \text{ is } A_1, \dots, x_k \text{ is } A_k) \ \textbf{THEN } y = g(x_1, \dots, x_k) \quad (7)$$

Where

i The numbers of the rules, in this case $i = (1, 2, 3, \dots, 15)$.

y The output variable.

$x_1 - x_k$ Variables inputs of the premise.

$A_1 - A_k$ Fuzzy sets whit linear membership functions representing a fuzzy subspace in which the implication R can be applied for reasoning.

f Logical functions that connects the propositions.

g Function that resolve the implication when input satisfies the premises.

Fuzzy sets of input, output and the set of rules are defined based on prior knowledge of system behavior. It also defines the linguistic variables and the type of fuzzy subsets [2] to be used. Therefore, it is necessary to define the type of input and output subsets.

Data provided by (5) corresponds to the inclination angle $(x = \theta)$. Using (9) the angular velocity of the inclination is obtained.

$$w = \Delta\theta \quad (8)$$

$$\Delta\theta = \widehat{w}_{k+1} - \widehat{w}_k \quad (9)$$

The angle and angular velocity of the robot are the fuzzy inputs, these indicate the stability of the robot, and the output represents the corrections to be made by servomotors, this is an angular increment, in order to keep the robot stable.

Figure 7 shows the fuzzy sets of inputs, θ with tilt angle input and w with angular velocity. The sensor angle is from 0º to 180º. However, in the ADC of the processor gives a value from 0 to 400, which allows working with more accurate data. In tests it has been observed that sharp changes in tilt angle speed can vary from -50 to 50 and it takes the safety range from -60 to 60.

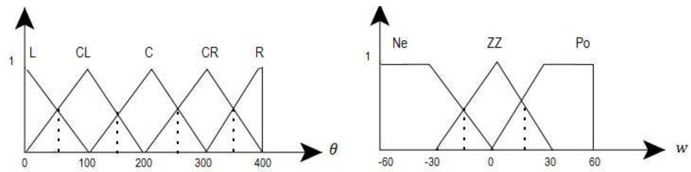

Figure 7. Input Fuzzy subsets.

The θ subsets membership are triangular, which is a trapezoidal set type. The subsets w have two trapezoidal membership and a triangular central one, this is very used format for membership functions [3]. Thus, (7) has two inputs $x_1 = \theta$ and $x_2 = w$. The subsets inputs above are *L, CL, C, CR, R,* defined for the linguistic variables *Left, Center Left, Center, Center Right* and *Right* respectively in θ and *Ne, ZZ, Po* to define for *Negative, Zero* and *Positive* respectively in w.

A TS characteristic of the first order model is that output functions are constant [2]. In (7) we have $y = g(x_1, \dots, x_k)$, where function g is a function as (10) [14], [16] and depends of the input variable and the i^{th} rule.

$$g(x, r) = p_{0r} + p_{1r}x_1 + \dots + p_{kr}x_k \quad (10)$$

In (10) is represented the consequent function, if coefficient p_{0r} is different to zero and the rest are equal zero, this method is called *singleton model* and this is a kind of Mandami Fuzzy control where the consequent of the rule is a singleton function instead of typical triangular o trapezoidal function [3], [14], that is

$$g(x, r) = p_{0r} \quad (11)$$

Using this strategy the output of the system will be:

$$y = \frac{\sum_R u_{R,a1,\dots,an} \cdot y_R}{\sum_R u_{R,a1,\dots,an}} \quad (12)$$

The $y_R = (p_{01}, \dots, p_{0n})$ values can be the subset given by the rules, where $(p_{01}, \dots, p_{0n})$ are the consequent singleton functions.

With this consideration, in Fig. 8 we have the output set whit *NB, N, Z, P* and *PB* defined for the linguistic variables *Negative B, Negative, Zero, Positive* and *Positive B* respectively.

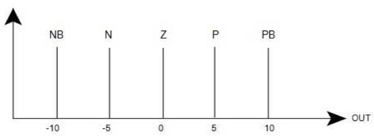

Figure 8. Output function set.

In the TS fuzzy design, is important the choice of the conjunctive operator, so, the connective **AND** between input variables can be **MIN** or **PROD**. The **MIN** and **PROD** conjunctive operators are often used just for simplification reasons [17]. **MIN** is used in practical cases and **PROD** is used in stability studies. Therefore, we used **MIN** for solving the connective **AND** because this research studies a practical case.

With these bases TS model was built by using:

- Connectivity **AND**: **MIN**.
- Implication: TS.
- Aggregation: **MAX**.

Therefore, considering (7), (10), (11) and the TS model bases the equation for rules will be:

$$R^i\text{: } \mathbf{IF}\ \theta\ is\ A_k\ \mathbf{AND}\ \ w\ \ is\ B_l\ \ \mathbf{THEN}\ Out\ \ is\ \delta_m \qquad (13)$$

Where θ and w are the system inputs, A_k and B_l are the subsets inputs respectively, Out is Out and δ_m is the output singleton subset.

The set of rules experimentally obtained, will be:

R1	*If*	θ is	L	*and*	w is	Ne	*Then*	Out is	***PB***
R2	*If*	θ is	L	*and*	w is	ZZ	*Then*	Out is	***P***
R3	*If*	θ is	L	*and*	w is	Po	*Then*	Out is	***P***
R4	*If*	θ is	CL	*and*	w is	Ne	*Then*	Out is	***P***
R5	*If*	θ is	CL	*and*	w is	ZZ	*Then*	Out is	***P***
R6	*If*	θ is	CL	*and*	w is	Po	*Then*	Out is	***Z***
R7	*If*	θ is	C	*and*	w is	Ne	*Then*	Out is	***P***
R8	*If*	θ is	C	*and*	w is	ZZ	*Then*	Out is	***Z***
R9	*If*	θ is	C	*and*	w is	Po	*Then*	Out is	***N***
R10	*If*	θ is	CR	*and*	w is	Ne	*Then*	Out is	***Z***
R11	*If*	θ is	CR	*and*	w is	ZZ	*Then*	Out is	***N***
R12	*If*	θ is	CR	*and*	w is	Po	*Then*	Out is	***N***
R13	*If*	θ is	R	*and*	w is	Ne	*Then*	Out is	***N***
R14	*If*	θ is	R	*and*	w is	ZZ	*Then*	Out is	**N**
R15	*If*	θ is	R	*and*	w is	Po	*Then*	Out is	**NB**

The system has 15 rules, then, is a "completeness" system, because for every combination of inputs there is a rule [16]. Under the state space analysis concluded that the switching line correctly separates positive and negative outputs, and therefore the system is stable, [18] as it is shown in Fig. 9.

But it has not been analyzed the complete stability of the system or if any of the rules is unnecessary, this is important because if a rule is not used may slow down the system.

This system has been pre analyzed using XFuzzy tools resulting in the following output surfaces in Fig. 10. It shows that the output can vary from 10 to -10, being 10 when the θ input value is 0, this is completely to the left and input w is in maximum angular velocity or minimum, this indicates that the control system is correcting the error as much as possible.

This final angular control is applied to the servomotors M4 and M5 of the robot to balance the force and avoid it falls.

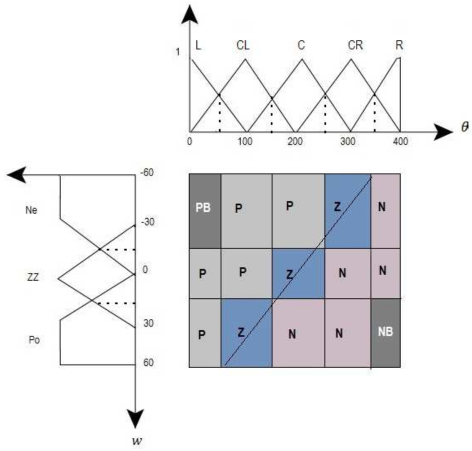

Figure 9. Line switching state space.

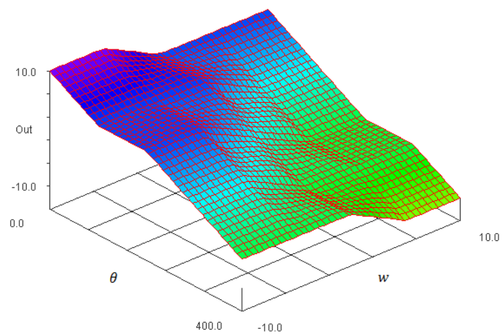

Figure 10. 3D Output Surface.

A MIMO system will be implement to compare the control results, if it are good the current control will be replaced for MIMO Fuzzy control, because with a single input the processor will work better [16].

The location of motors can be observed in Fig. 11.

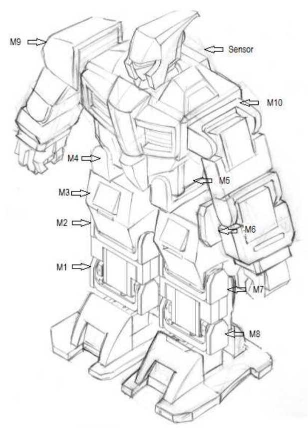

Figure 11. Motor location and Human Robot.

The control gives good results, how it can be seen in Fig. 12 and Fig.13. In the first, there are three signals (angle, angular velocity and the angular correction). The last corrects the stabilization of the robot around 200 in angle and zero in angular velocity. Finally in Fig.13 the linguistic trajectory shows the transitions of the variables through the rules map. In this case starts in *R12* then following *R11*, *R10*, *R8*, *R9* and stop in R8 where the variables are near to goal point.

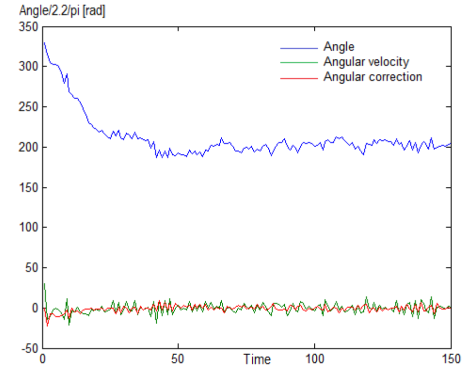

Figure 12. Input signals and resulting angular correction.

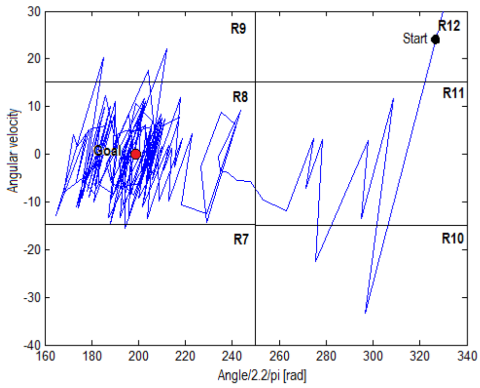

Figure 13. Linguistic trajectories.

IV. ACKNOWLEDGMENTS

This work was supported by Senescyt Ecuador. This work was partially financed by Comunidad de Madrid through the project Robocity 2030 II - CM (CAM – S2009/DPI-1559)

V. CONCLUSION

The balance fuzzy control allows the robot to achieve the objective of reproducing tele operated movements without falling. The Kalman filter reduces Gaussian noise produced by the sensor. This is easily calibrated, depending on the standard deviation of the system. This makes feasible the use of a fuzzy controller. This case has 2 inputs and one output. The inputs, angle and angular velocity, take into account not only if the robot tends to fall but at what speed, making the system more robust and the firsts obtained results promises a good ending.

The fuzzy method applied to this problem deals a low processing cost. Using a low computational processor forces the developer to optimize all resources, because a single processor is responsible for controlling the motors, do the Kalman filtering, do the Fuzzy control and make balance adjustment.

The operation delay is around 100 ms. This time will be reduced by replacing the processor with a higher performance and faster math. We propose to implement the control in two axes, using the method of inverted pendulum, and to obtain differential equations that allow the robot to walk more naturally. In addition, we plan to obtain rules and membership functions of the fuzzy controller in an automatic way.

REFERENCES

[1] R. Vullings, B. de Vries, and J. W. M. Bergmans, “An Adaptive Kalman Filter for ECG Signal Enhancement”, *Biomedical Engineering, IEEE Transactions on*, vol. 58, n o. 4, pp. 1094 –1103, abr. 2011.

[2] K. Michels, “Fuzzy Control, Stability and Design of Fuzzy Controllers”, J. Kacprzyk, Ed. Poland: Springer, 2006, pp. 10-42.

[3] R. Babuska, “Fuzzy Modeling for Control”, H. J. Zimmermann, Ed. Boston, London: Kluwer Academic Publisher, 1998, pp 245-256.

[4] T. Liu, Y. Inoue, and K. Shibata, “A Wearable Sensor System for Human Motion Analysis and Humanoid Robot Control”, in *Robotics and Biomimetics, 2006. ROBIO ’06. IEEE International Conference on*, 2006, pp. 43 –48.

[5] A. Alvarez-Alvarez, G. Trivino, and O. Cordon, “Human Gait Modeling Using a Genetic Fuzzy Finite State Machine”, *Fuzzy Systems, IEEE Transactions on*, vol. 20, n o. 2, pp. 205 –223, abr. 2012.

[6] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi, “Imitating human dance motions through motion structure analysis”, in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, vol. 3, pp. 2539 – 2544 vol.3.

[7] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, “Whole body humanoid control from human motion descriptors”, in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 2677 –2684.

[8] E. Foxlin, “Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter”, in *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, 1996, pp. 185 –194, 267

[9] L. Tamas, G. Lazea, R. Robotin, C. Marcu, S. Herle, and Z. Szekely, “State estimation based on Kalman filtering techniques in navigation”, in *Automation, Quality and Testing, Robotics, 2008. AQTR 2008. IEEE International Conference on*, 2008, vol. 2, pp. 147 –152.

[10] L. W. Stimac and T. A. Kennedy, “Sensor alignment Kalman filters for inertial stabilization systems”, in *Position Location and Navigation Symposium, 1992. Record. 500 Years After Columbus -Navigation Challenges of Tomorrow. IEEE PLANS ’92., IEEE*, 1992, pp. 321 –334.

[11] C. Hide, T. Moore, and M. Smith, “Adaptive Kalman filtering algorithms for integrating GPS and low cost INS”, in *Position Location and Navigation Symposium, 2004. PLANS 2004*, 2004, pp. 227 – 233.

[12] Kalman, R. Emil, “A New Approach to Linear Filtering and Prediction Problems” *ASME Trans. Journal of Basic Engineering* vol. 82 D, 1960, pp 35-45

[13] International Electrotechnical Commission (IEC), “Programmable Controllers”, IEC 1131. Part 7: *Fuzzy Control Programming*. IEC 1131-7, Jan. 1997.

[14] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and Its Applications to Modeling and Control”, *in Systems, Man and Cybernetics, IEEE Transactions*, 1985, vol. SMC-15, No. 1, pp116 -132.

[15] S. F. Desouky and H. M. Schwartz, “Genetic based fuzzy logic controller for a wall-following mobile robot”, in *American Control Conference, 2009. ACC ’09.*, 2009, pp. 3555 –3560.

[16] Kevin M. Passino and Stephen Yurkovich, “Fuzzy Control”, Department of Electrical Engineering, The Ohio University, Addison-Wesley Longman, Inc, 1998, pp 12 - 77.

[17] M. Benrejeb, A. Sakly, D. Soudani, and P. Borne, “Stability Domain Study of Discrete TSK Fuzzy Systems”, in *Computational Engineering in Systems Applications, IMACS Multiconference on*, 2006, vol. 1, pp. 293 –298.

[18] Li-Xin Wang, “Adaptive Fuzzy Systems and Control Design and Stability Analysis”, University of California at Berkeley, PTR Prentice Hall, 1993, pp 300-320.