International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

Artificial Heavy-Duty Structural Technology For Ai Mobile Robots To Control Dynamic Programming

Dr.Syed Khasim[1], Dr. M.S. Rohokale [2], Sathish Kumar D[3], B Sarath Chandra[4], Dr.Shaik Shakeer Basha[5], Rahul Neware[6]

[1]Professor, Department of Computer Science and Engineering, Dr.Samuel George Institute of Engineering & Technology, Markapur, Prakasam Dt, Andhra Pradesh-523316.
[2]Professor, Mechanical Engineering, SKN Sinhgad Institute of Technology and Science Kusgaon (Bk) Lonavala Pune Maharashtra-410401.
[3]Assistant Professor (Sr.G), Department of EEE, KPR Institute of Engineering and Technology, Coimbatore-641407.
[4]Assistant Professor, Department of Computer Science and Engineering,PSCMR College of Engineering and Technology, Vijayawada, Andhra Pradesh 520001.
[5]Assistant Professor, Department of Computer Science and Engineering, Avanthi Institute of Engineering and Technology, Gunthapally, Abdullahpurmet Mandal-501512, Hyderabad, Telangana.
[6]PhD Research Fellow, Department of Computing, Mathematics and Physics , Høgskulen på Vestlandet, Inndalsveien 28, 5063 Bergen, Norway.

Abstract

This article offers an intelligent four-wheeled mobile robot movement controller with four omni directional drive wheels equally spaced by each other at 90^o using a Field-Programmable (FPGA) with artificial immune system algorithm. The suggested FPGA-based Artificial Intelligent System (AIS) method combines machine learning with FPGA technology. These FPGA-based AIS auto-stunning intelligent controls are projected to outperform standard non-optimized controls, the dynamic programming controllers and the particle swarm controller.

Keywords: Mobile robots; Artificial Immune System; using field-programmable gate array

1. INTRODUCTION

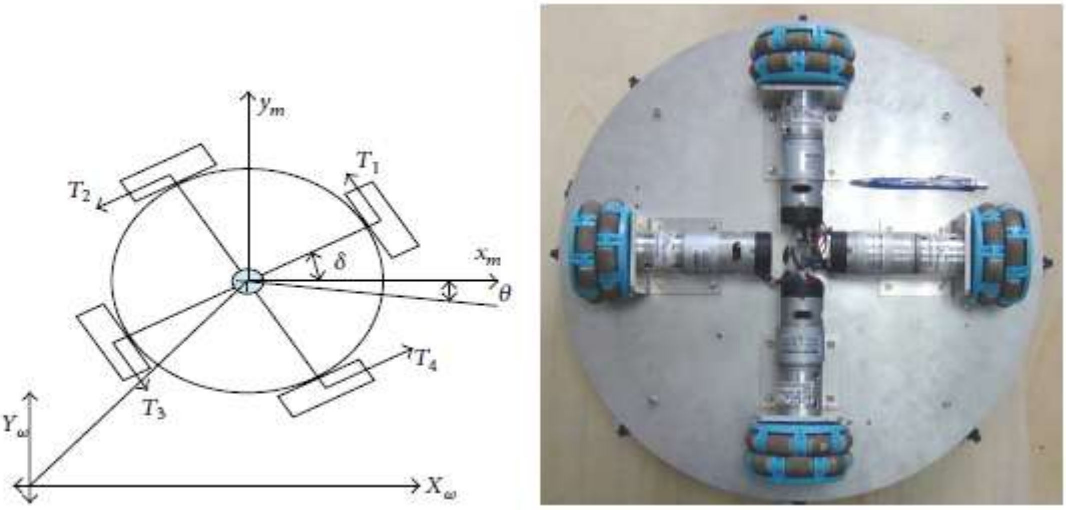
In terms of locomotion mechanism, these robots outperform those using differential wheels [1]. Byun et al. [2] built a mobile robot with a changeable wheel movement in particular. Purwin [3] suggested a four-wheeled omnidirectional vehicle trajectory generating system. For an omnidirectional mobile robot, it was suggested a fuzzy route proposed a fuzzy controller for an directional inspection machine [4].

Between these contemporary metaheuristic-based solutions to address classification problem in mobile robots, de Castro and Timmis [5] introduced the AIS algorithm, which is based on the biological immune system and has proven to be a effectual and computational example for NP-hard combinatorial method. Its paradigm is based on the immune system's natural reaction. By leveraging their high optimization ability, the adaptive and AIS method been effectively utilized in a variety of fields, including machine learning, classification, and pattern recognition.

FPGA invention has ushered in a fundamental shift in integrated circuit design for AI computing. In various fields, such as fuzzy positioning algorithm, it has been demonstrated to be an advanced and productive way of realizing complex algorithms. This FPGA technology has indeed been demonstrated to be effective in designing computational intelligence in embedded systems that incorporate memory, and processor cores, with the advantages of adaptability, hardware/software co-design, and copyright recyclability. However, no attempt has been made to build a mobile robot-specific intelligent FPGA-based AIS controller. The work is to develop practical algorithm controller based on FPGA integrated AIS for autonomous vehicles that can monitor and stabilize their path.

2. KINEMATIC CONTROL

This report explains the kinematics of an directional robotic system with wheels separated at 90^{o} each other. The integrated controller is presented to accomplish stabilization and path tracking using the kinematics. In relation to a global frame, Figure 1 illustrates the set of driving arrangement.



(a) (b)

Figure 1: Arrangement of four-wheeled robot.

$$\upsilon(t) = \begin{bmatrix} \upsilon_1(t) \\ \upsilon_2(t) \\ \upsilon_3(t) \\ \upsilon_4(t) \end{bmatrix} = \begin{bmatrix} r\omega_1(t) \\ r\omega_2(t) \\ r\omega_3(t) \\ r\omega_4(t) \end{bmatrix} = P(\theta(t)) \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix}, \qquad (1)$$

where

$$P(\theta(t)) = \begin{bmatrix} -\sin(\delta+\theta) & \cos(\delta+\theta) & L \\ -\cos(\delta+\theta) & -\sin(\delta+\theta) & L \\ \sin(\delta+\theta) & -\cos(\delta+\theta) & L \\ \cos(\delta+\theta) & \sin(\delta+\theta) & L \end{bmatrix}, \qquad (2)$$

where r signifies the wheel radius; L denotes the space between the wheel's centre and the geometric centre of the movable platform; The velocities of wheel was denoted by i(t) and i(t), with I = 1, 2, 3, 4, etc. The mobile robot's posture is represented by [x(t) y(t) (t)].

$$P^{\#}(\theta(t)) = \begin{bmatrix} \frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2} & \frac{\cos(\delta+\theta)}{2} \\ \frac{\cos(\delta+\theta)}{2} & \frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2} \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix}. \quad (3)$$

This part uses the kinematic model in (1) to develop kinematic controls for the directional mobile robot in Figure 2 to accomplish point stabilization and trajectory tracking. To create the controller for motion, one must first specify error, the difference between the current and point position.

$$Z_e(t) = \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} - \begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} \quad (4)$$

which gives

$$\begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = P^{\#}(\theta(t)) \begin{bmatrix} r\omega_1(t) \\ r\omega_2(t) \\ r\omega_3(t) \\ r\omega_4(t) \end{bmatrix}. \quad (5)$$

$$\begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} = \frac{1}{r} P(\theta(t)) \times \left(-K_P \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau)\,d\tau \\ \int_0^t y_e(\tau)\,d\tau \\ \int_0^t \theta_e(\tau)\,d\tau \end{bmatrix} \right). \quad (6)$$

The PID control is extensively used in industrial applications [6], is employed in the control system presented in (6). When (6) is added to (5), the closed system error system's behaviors become

$$\begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = -K_P \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau)\,d\tau \\ \int_0^t y_e(\tau)\,d\tau \\ \int_0^t \theta_e(\tau)\,d\tau \end{bmatrix}. \quad (7)$$

$$V_1(t) = \frac{1}{2}\left[x_e(t)\ \ y_e(t)\ \ \theta_e(t)\right]\begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} + \frac{1}{2}\left[\int_0^t x_e(\tau)\,d\tau \ \ \int_0^t y_e(\tau)\,d\tau \ \ \int_0^t \theta_e(\tau)\,d\tau\right] K_I \times \begin{bmatrix} \int_0^t x_e(\tau)\,d\tau \\ \int_0^t y_e(\tau)\,d\tau \\ \int_0^t \theta_e(\tau)\,d\tau \end{bmatrix}. \tag{8}$$

Taking the time derivative of $V_1(t)$, one obtains

$$\dot{V}_1(t) = \left[x_e(t)\ \ y_e(t)\ \ \theta_e(t)\right]\begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} + \left[\int_0^t x_e(\tau)\,d\tau \ \ \int_0^t y_e(\tau)\,d\tau \ \ \int_0^t \theta_e(\tau)\,d\tau\right] K_I \times \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} = -\left[x_e(t)\ \ y_e(t)\ \ \theta_e(t)\right] K_P \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} < 0. \tag{9}$$

3. AIS ALGORITHM

In order to discover better solutions to complex combination problems, AIS algorithms employ an immune system and hypermutation. This methodology offers many strategies to address actual issues when conventional methods do not work, comparable to bio-inspired metaheuristics from GA [7] and PSO [8]. In order to test its efficacy, each antikörper is carefully preset with the optimization approach. The AIS-affinity anticorps are cloned, hypermutated and selected while the population is also increased with arbitrary anticorps. Worsening antibodies are substituted by superior mutation clones. This group is developed until the end is satisfied.

The steps in the AIS algorithm follows.
Step 1: Set up the AIS and iteration count.
Step 2: Create antibodies community of size (Ab1,..., Abs) at random.
Step 3: Using affinities tool, compute the affinity ratio for each antibody.
Step 4: Create a group set for effective antibodies. The length of the replica is determined by the number of duplicates: the greater the affinity, the bigger the replica.

Step 5: Refresh your antibody.
Step 6: Use the hyper mutation process to expand the variety of the clone set and create the developed cloning group.
Step 7: Examine the criterion for stopping.

An antigen is a chain of integral variables of the AIS method consisting of parameters for optimization problems. It should be noted that the beginning number of each antibody is randomly generated. This coding method is all similar to optimization computation, PSOs, and GAs. The notion of space-form was created to measure and resulting in basic conclusions on interactions between receptors and antigenes. The length can be represented in euclidean as the Ag-Ab binding vectors (14).

$$D = \sqrt{\sum_{i=1}^{L} (\mathrm{Ab}_i - \mathrm{Ag}_i)^2}. \qquad (14)$$

The chosen antibodies are then cloned in order to increase their affinity for the invading antigens.
When a B cell is activated, a hypermutation mechanism in the cell's variable area is initiated, according to the suggested AIS.
The mechanism is essential for the development of different antibody receptors as well as the improvement of antibody sensitivity and selectivity.

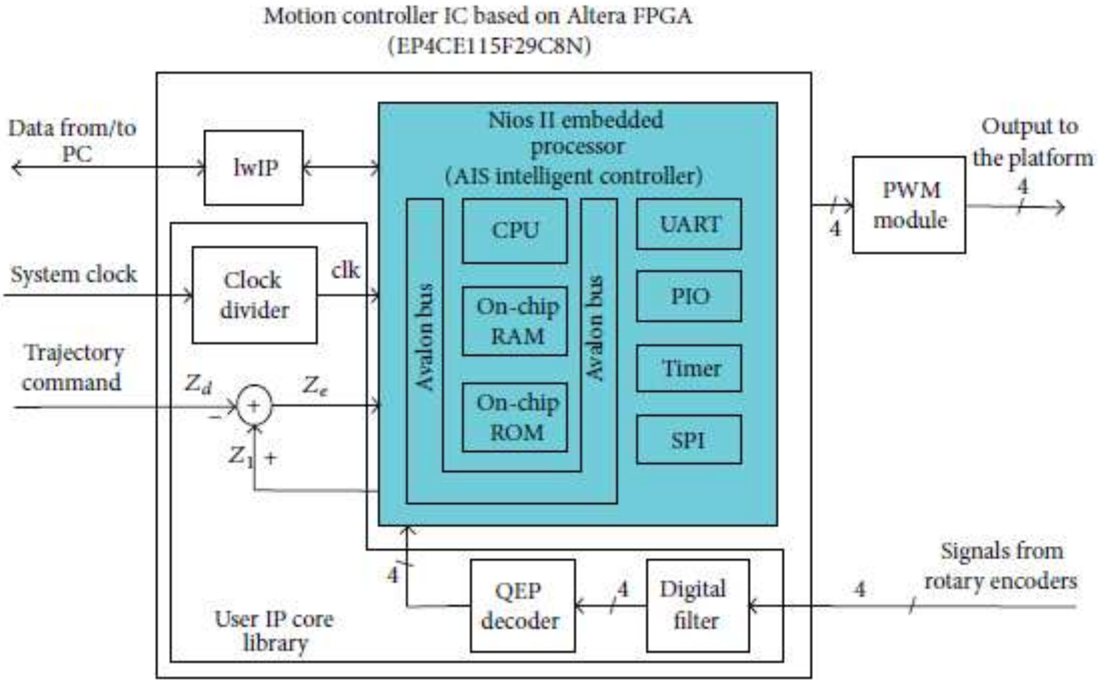


Figure 2: FPGA with AIS

4. CONCLUSIONS

This paper proposes a smart motion control system using FPGA/AIS computing capability for monitoring and stabilizing the course of a robot manipulator directionally with 4 distinct 90° drive rotors. The AIS computer architecture built on the cinematic movement model was converted to an intelligent kinematic actuator. The AIS parameter tuner and the film movement controller are integrated on one FPGA Chip to efficiently produce a functional robotic system.

5. REFERENCES

[1] T. Kalmar-Nagy, R. D Andrea, and P. Ganguly, “Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle,” Robotics and Autonomous Systems, vol. 46, no.
1, pp. 47–64, 2004.

[2] K.-S. Byun, S.-J. Kim, and J.-B. Song, “Design of a four-wheeled omnidirectional mobile robot with variable wheel arrangement mechanism,” in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 720–725, Washington, DC, USA, May 2002.

[3] Dr.P.Sivakumar. “Trailing Mobile Sinks to Data Coverage Protocol for Wireless Sensor Networks”, International Journal of Innovations & Advancement in Computer Science (IJIACS), Vol.5, Issue No: 6, June 2016.

[4] C.-C. Shing, P.-L. Hsu, and S.-S. Yen, “T-S fuzzy path controller design for the omnidirectional mobile robot,” in Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics (IECON ’06), pp. 4142–4147,November 2006.

[5] Sridharan, K., & Sivakumar, P. (2018). A systematic review on techniques of feature selection and classification for text mining. International Journal of Business Information Systems, 28(4), 504-518.

[6] D. H. Kim, “Tuning of 2-DOF PID controller by immune algorithm,” in Proceedings ofthe 2002 Congress on Evolutionary Computation, vol. 1, pp. 675–680, 2002.

[7] J. C. Gallagher, S. Vigraham, and G. Kramer, “A family of compact genetic algorithms for intrinsic evolvable hardware,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 2, pp. 111–126, 2004.

[8] Latchoumi, T. P., Balamurugan, K., Dinesh, K., & Ezhilarasi, T. P. (2019). Particle swarm optimization approach for waterjet cavitation peening. Measurement, 141, 184-189.