Iraqi Journal of Industrial Research (IJOIR)

Journal homepage: http://ijoir.gov.iq

A Novel HSPICS for Industrial Robotic Controller Based on FPGA_SoC: Modelling and Fabrication

[1]Falih Salih Mahdi Alkhafaji*, [2]Wan Zuha Wan Hasan

[1]State Company for Electrical and Electronic Industries, Ministry of Industry and Mineral – Iraq

[2]Institute of Plantation Studies, Faculty of Engineering, UPM University – Malaysia

Article information

Article history:
Received: May, 21, 2022
Accepted: August, 26, 2022
Available online: October, 20, 2022

Keywords:
HSPICS,
HSMDAQ,
GA,
FPGA_SoC,
Robotic Controller

Corresponding Author:
Falih Salih Mahdi Alkhafaji
falih2009@gmail.com

DOI:
https://doi.org/10.53523/ijoirVol9I2ID172

Abstract

Tremendous studies have been proposed to optimize PI controller based Genetic Algorithm (GA) to improve the speed performance of DC motor commonly required in robotic applications. In PID controller, there are very few studies to overcome the drawbacks of classical GA, besides little pay attention to improving the speed performance of a DC motor to be measured in the microsecond unit. The main target is to maximize reduction step response characteristics, by proposing to design and fabricate a high speed proportional integral controller system (HSPICS). The primary methodology includes three sub methodologies using several new techniques and procedures to achieve three objectives. Firstly, to propose an improved genetic algorithm (IGA) to enhance the performance for better searching constraints for PI controller. Secondly, generate VHDL based Simulink model without needing expensive software. Finally, integrate the proposed controller-based on FPGA_SoC using Embedded Coder and FPGA in the loop (FIL) techniques to run the design based model. To show the effectiveness of the proposal, it was used three different DC motors. Simulation results show that the proposed controller achieves much higher reduction step response ratios (RSRR) compared with classical GA and PSO, further shortened step response characteristics to be measured in the microsecond unit. Analyzing the performance demonstrates that the RSRR has been enhanced for motors 1, 2, and 3 by 8, 9, and 35 times over classical GA, and 3, 3, and 10 over PSO, respectively. The comparison response time results between simulation and experimental for the studied motors show that the steady state time ratios (SST) were minimized significantly.

1. Introduction

Nowadays, Industrial robot offers a best solution to improve production efficiency, quality, and much more benefits in manufacturing process [1]. The robotic technology have been increasing interest in the development of controllers for their custom design [2]. The DC motors have been used in robotic applications in case of their simplicity, ease of use, reliability, and cost effective [3]. The chosen controllers play an essential role in the design of any plant system. PI and PID algorithms have been widely in DC motor controller due to easy tuning, robust,

effectiveness, and can be realized easily in engineering[4-6]. In fact, PID controller is unique for each application, and some implementations may use only one or two parameters such as PI, PD, P, I [7, 8]. Control system performance is often estimated by applying a step signal to evaluate the step response characteristics in terms of dead time (td), rise time (tr), settling time (ts), and peak overshoot [9, 10]. By contrast, tuning proportional gains of the controller should be appropriately tuned to obtain a desired closed-loop system performance [11-13]. Previously, there are massive tuning methods have been proposed but insufficient to obtain satisfied proportional gains in case of poor damping, poor robustness, and inappropriate for a second-order system [14-17]. Consequently, evolutionary optimization algorithms have been invented to get a better solution over than classical tuning methods. GAs and PSOs are the most evolutionary algorithms that were widely used in industrial applications, offers a capillarity for handling issues with non-linear constraints, multiple objectives, and dynamic components properties [18-20]. Based on a survey, classical GA is not a best solution with respect to PSO. Profoundly negative side in GA comes from the way to formulate the new generation after the initial generation, contains some random components, which leads to corrupt generation values in the primary stages of global searching, that may affect the negative side for searching constraints [21-25]. Additionally, one of the most crucial problems in genetic PI and PID controllers cannot get satisfactory results in case of the difficulties of the chosen the best GA parameters and operators. Currently, there is no precise algorithm can get a best solution to optimize the PI controller for speed DC motor [26-29].

On the one side, parameters of DC motor vary with time due to the depreciation and aging effect which reduces performance. Thus, most industrial processes are non-linear, and some process is challenging to establish the estimation of TF form. Consequently, the classical PID controller cannot achieve desired closed-loop response if the mathematical form of a TF of such plant system not modelled accurately [30, 31], thus, without achieving an accurate TF, the implementation of compensator design and stability analysis is rendered insufficient [32, 33], and this is the second challenge to get higher performance controller design . There are several studies for estimation TF used digital devices such as oscilloscope to collect data in data out to be imported into system identification application (Sys Ident).However, there is a drawback in this method that comes from lousy collection data with limited sampling, causing the lowest fitting between injected signal and angular speed (θ), leads to lousy estimation TF form [34, 35]. Other studies used data acquisition instruments to collect data, but not focuses on how to minimize the noise which comes from the encoder sensor, which leads to reduce the accuracy of the estimation [36, 37]. The third challenge in designing controller is that the low engagement between simulation and experimental design and how to reduce the deviation between them. Hence, integrating controller based hardware circuitry was considered a very crucial point, especially for precise applications such as the controller based optimization algorithms ,which depends on the efficiency of hardware selection [38-41].

Therefore, to get high performance real-time control system, designers need to select the right choice between three main families of digital device technologies: (1) ASIC could be in charge of reducing errors, compensation of interferences with reducing area and power efficiency, but it is quite rigid, and their programmability is low; (2) Microcontrollers could be used to implement hardware circuitry, but it consumes areas of design with more complexity, (3) Field programmable gate arrays (FPGA)is considered the best solution for hardware implementation [42-48], in case of high flexibility, reprogrammable, rising integration scale, and large scale integration [49, 51] . Novel advanced features embedded in this device bring a new scenario to be used for modern applications [52]. However, this technology needs more time to learn how to integrate algorithm circuitry, besides needing expensive software to generate VHDL or Verilog language [53, 54]. To engage the maximum performance for designing robotic controller, several criteria should be accurately evaluated in simulation parts such as the accuracy of TF form, besides studying a new modification to improve GA performance in case of the basic operating mode in classical GA is not always well suited to any constraint treatment [55] . The aim is to design high speed controller for a robotic applications to accomplish highest reduction response time to be measured experimentally in microsecond unit, where the most significant related works were obtained the results in millisecond unit [56-68], further decreasing the deviation of between experiment and simulation, and reducing the average error step response (AESR) bellow 10%. The contribution of this work could be concluded in four points: (1) Improve GA(IGA) based PI controller to engage maximum performance to be used in wide range applications. (2) Identifying the TF form of any plant system without needing manufactories parameters. (3) Significantly improves the accuracy of PI controller. (4) Generate VHDL based Simulink model without needing expensive

software. (5) Integrate the proposed controller on FPGA_SoC offering high speed performance circuitry, low consuming area of design, and high accuracy.

The rest of this paper is organized as follows: Section 2 explains the simulation and experimental methodology of the proposed HSPICS based IGA, the experimental procedure divided into four parts to implement the hardware circuitry of the proposed, besides presents the effectiveness of the hardware platform based FPGA_SoC. The simulation and experiment results are described and discussed in section 3, showing how the proposed IGA algorithm influence to improve the PI controller.GA and PSO have been carried out as a benchmark comparison to evaluate the performance of the proposed algorithm, further discussed the simulation and experimental results, followed by the conclusion and recommendation in Section 4.

2. Simulation and Experimental Procedure

As shown in Figure (1), the research's strategy relies on three sub methodologies to design HSPICS as follows: (1) using high speed motor data acquisition HSMDAQ system to estimate accurate TF forms of three different types DC motors (56RPM,4000RPM,107RPM), all phases were discussed in [69] , and to measure the step response of the proposed controller system ; (2) proposed improved GA (IGA) based PI controller, it was used classical GA and PSO as a benchmark comparison. The comparison simulation results evaluated in terms of step response characteristics; (3) proposing a new method to generate VHDL code of proposed HSPICS and to fabricate high quality hardware on FPGA without needing Vivado's license. To evaluate the effectiveness of the design. To show the efficiency of the integrated hardware controller, the comparison between simulation results and experimental results was done in terms of td, tr, ts, and SST.

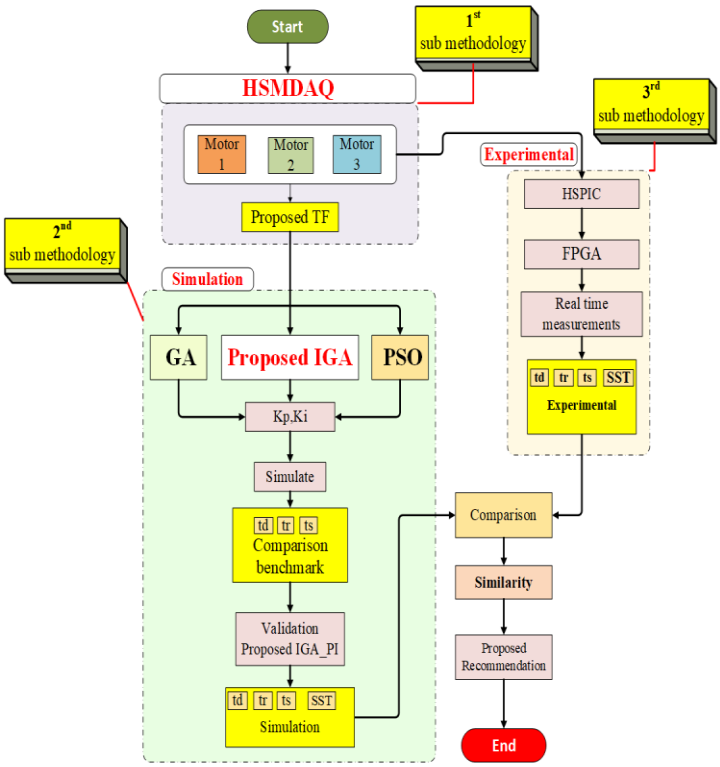

Figure (1). Methodology of design HSPICS.

A. *Proposed IGA based PI Controller*

As shown in Figure (2), to improve GA performance there are three optimizations levels were suggested as follows: (1) using MIFF to optimize proportional gains of PI controller as demonstrated in [70] , then added the resulted gains to initial random to be established in fitness function using PI controller expression; (2) imported the proposed fitness function MIFF into the GA toolbox to proceed the first new generation to be encoded and decoded the first chromosomes in the global population, all these phases were explained in [70, 71]; (3) proposed a new OGA_PO procedure as a third optimization level, by studying the effects of varying GA parameters and operators

sequentially to obtain better searching constraints in global optima. This procedure relies on modifying the following operators: (1) population size and variable boundaries; (2) fitness scale; (3) migration direction with fraction and interval values; (4) initial penalty and penalty factor; (5) generation values; (6) function tolerance.

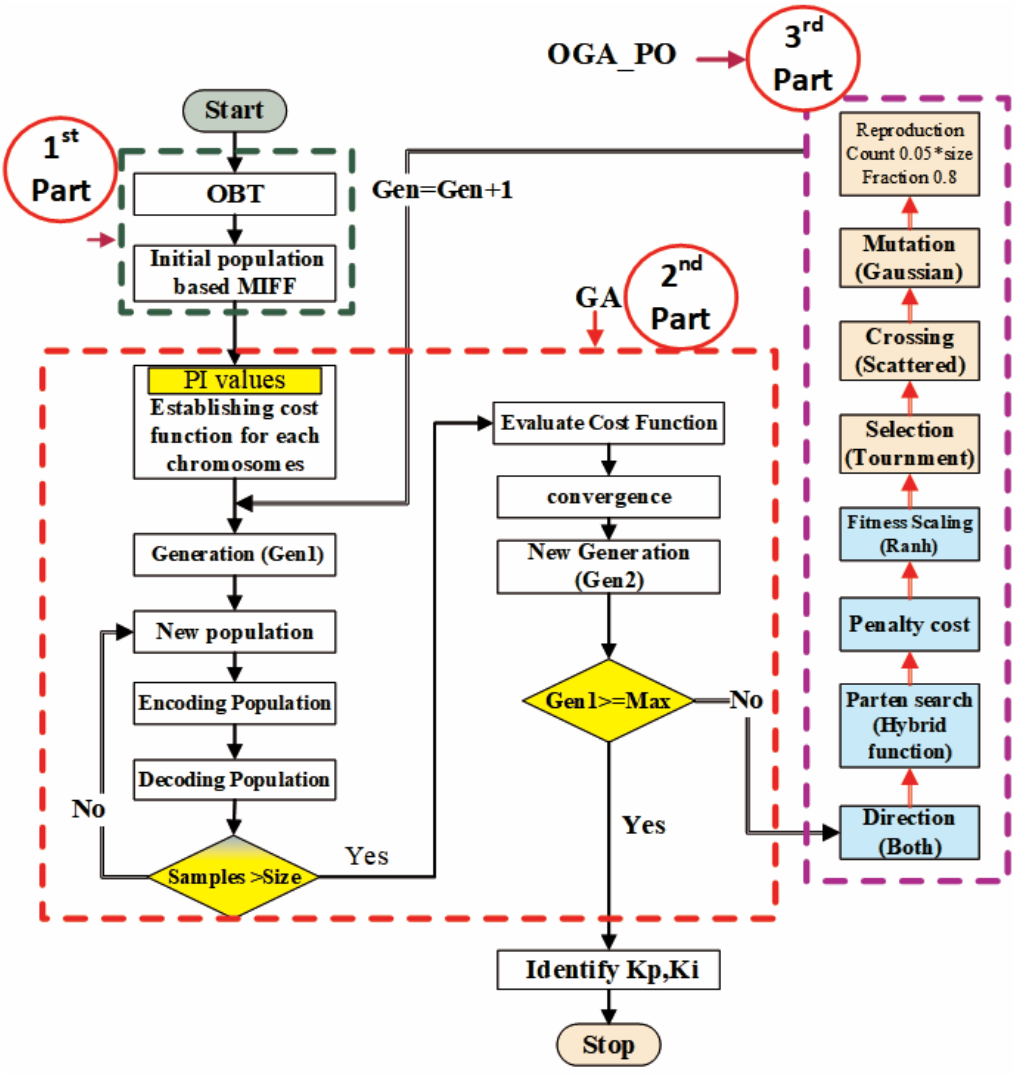

Figure (2). Proposed IGA based PI controller.

Based on third optimization level, Figure (3) shows the proposed OGA_PO procedure includes several steps to optimize GA parameters and operators as follows:

1. Import IAE fitness function-based MIFF for M1 into GA toolbox.
2. Setting GA parameters and operators-based MATLAB recommendation.
3. Setting Bound size (B) to 100.
4. Loop B=B+i, i=1 to 10, step 100
5. Measure Response (Ti) in terms of (td, tr, ts), if Ti<Ti+1, fixed B to 700
6. Loop Population Size (N), i=1 to 10, j=0 to 100, step 10, n=1 to 100
7. Generate chromosomes x1, x2, xn, based MIFF and measure Ti
8. Increment Ni=Ni+j, Generate chromosomes x1, x2,…,xn, measure Ti , goto step 6
9. Measure Ti, if Ti<Ti+1, fixed N to 50
10. Setting fitness scaling (FS): Rank, Shift Linear, Top, measure Ti, fixed to (Rank)
11. Setting Selection fitness (S): Tournament, Reminder, Roulette, Uniform, measure Ti, Fixed (Tournament).
12. Setting Mutation (M): Gaussian, Adaptive, Uniform, Constraint Dependent, Measure Ti, Fixed (Gaussian)
13. Setting Gaussian Scale (GS)=1

14. Loop, i=1 to 101, GS(i)=GS(i)+0.01, measure Ti, if Ti<Ti+1, Fixed GS=1.525
15. Loop, i=1 to 6, Setting Crossover (C): Scattered; single point; two points; intermediate, heuristic, arithmetic, Dose Ti<Ti+1, Fixed to Scattered.
16. Loop i-1 to 2, Setting Migration Direction (MD): Forward, Both, Dose Ti<Ti+1, Fixed to (Both).
17. Setting Migration Fraction (MF), MF (1) =0
18. Loop i=1 to 10, MF(i)=MF(i)+0.2, if Ti<Ti+1, Fixed MF=1
19. Setting Function Tolerance (FT)=1E-6
20. Setting Non-Constraint (NT)=1E-6
21. Loop i=1 to 5, FT(i)=FT(i)*E-i, Ti<Ti+1, Fixed FT=1E-8
22. Loop i=1 to 5, NT(i)=NT(i)*E-I, Ti <Ti+1, Fixed NT=1E-8
23. Loop i=1to 4, setting Hybrid Function: Pattern Search, Fminsearch, Fminnumc, Fmincon., if Ti<Ti+1, Fixed Pattern search
24. Repeat the same procedure for M2 and M3
25. End

The obtained optimal parameters and operators produces by OGA_PO illustrated in Table 1, which can be selected to enhance the performance level of GA for searching constraints. The best GA parameters achieved when using the Lower Band (LB) and Upper Band (UB) between of 0 to 700, with population size equals to 50 individuals in each generation. The best Gaussian Scaling (GS) in mutation function accomplished at 1.52 deviation to make small changes in the individuals in the population, which provides genetic diversity and enables GA to search broader space. The optimal nonlinear constraint parameters of the initial penalty and penalty are resulted by 10 and 100 respectively. On the other side, the best operators for Fitness Function (FS), Selection Fitness (S), Mutation (M), Crossover (C), Migration Direction (MD) and Hybrid Function (HF) achieved at Rank, Tournament, Gaussian, Scattered and Both directions, Pattern search respectively. The resulted parameters and operators can be applied to GA solver to provide a high accuracy searching in terms of encoding chromosomes, crossover selected gens from parent, and to create a new offspring and mutation offspring. Ultimately, the suggested strategies can produce a better optimal solution to improve GA performance significantly, leads to overcoming traditional GA and PSO in terms of step response characteristics.

Table (1). The Optimal Parameters and Operators**.**

Symbol	Parameter and Operator	Optimal Selection
LB	Lower Band	[0 0 0]
UB	Upper Band	[700 700 700]
FS	Fitness Scaling	Rank
S	Selection Fitness	Tournament
M	Mutation	Gaussian
GS	Gaussian Scaling	1.52
C	Crossover	Scattered
MD	Migration Direction	Both direction
MF	Migration Fraction	1
FT	Function Tolerance	1E-8
NT	Non Constraint Tolerance	1E-8
HF	Hybrid Function	Pattern search
CP	Initial penalty	10
PF	Penalty Factor	100
Itr.	Iteration	50
BS	Bound Size(B)	700
PS	Population Size	50

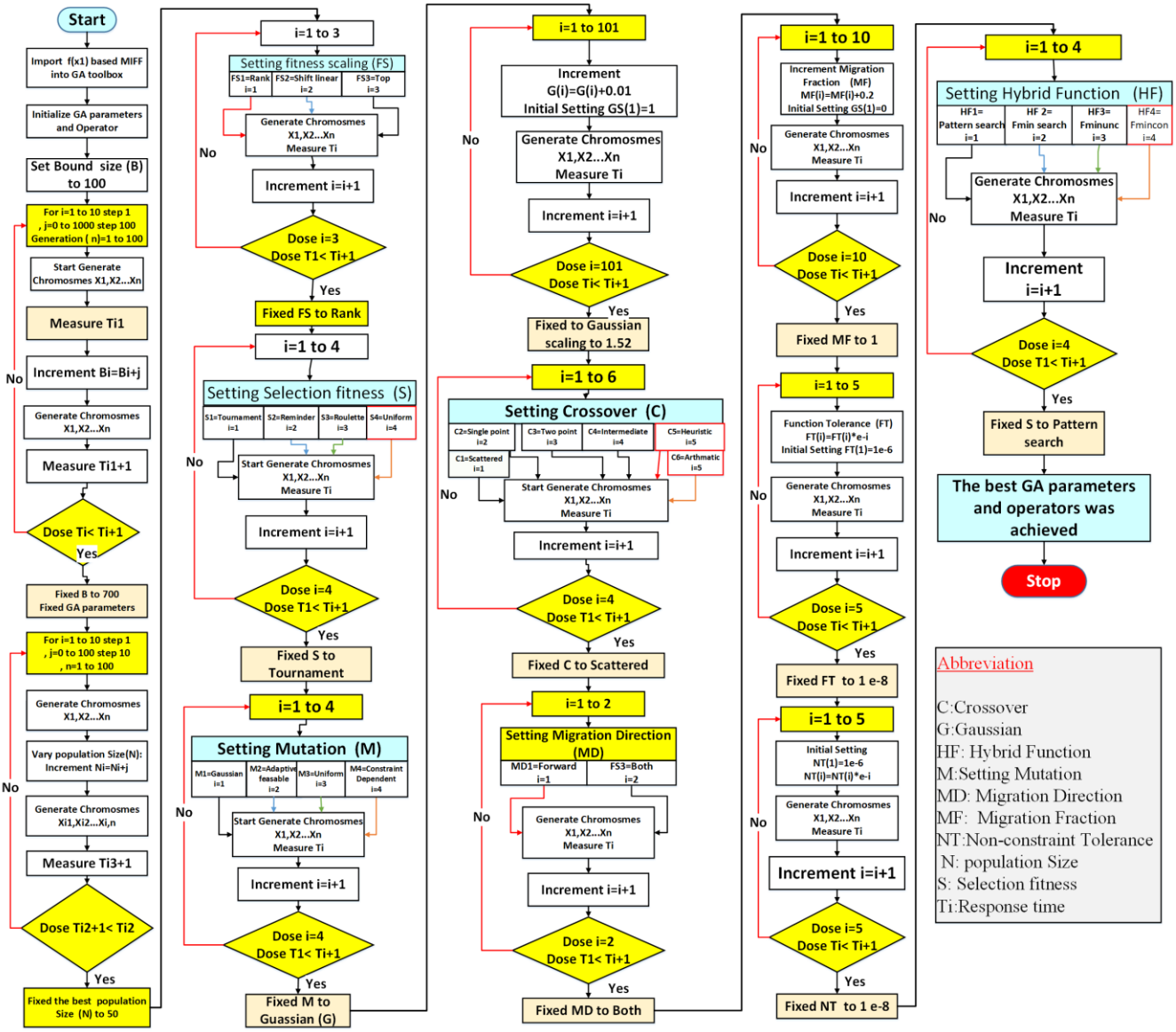

Figure (3). Proposed OGA_PO procedure.

B. *System Configuration*

The major problem in the implementation of hardware based FPGA is that cannot calculate a floating-point without external support code or fixed-point unit on board. Therefore, it was introduced a new methodology to find the best solution for signal conversion and fixed-point calculations. As shown in Figure (4), the methodology is divided into four parts providing time saving, accurate design, and improving the simplicity of the model. The first part using Embedded Coder application to generate the C and then to generate VHDL code for model based design, using Vivado Hls 2016.3 side by side with MATLAB.

Next to generate the bit stream and to fabricate the proposed controller within the HSMDAQ system to acquire data, and measure experimentally the step response characteristics using a scope simulator. To increase the accuracy, it was used a tachometer device to calibrate the measuring speed through a scope simulator. The third part is to run the proposed controller based Simulink model using the FIL technique to measure the simulation of the step response. The last part is to compare the step response characteristics between experimental and simulation results to show the deviations between them and hardware percentage error.

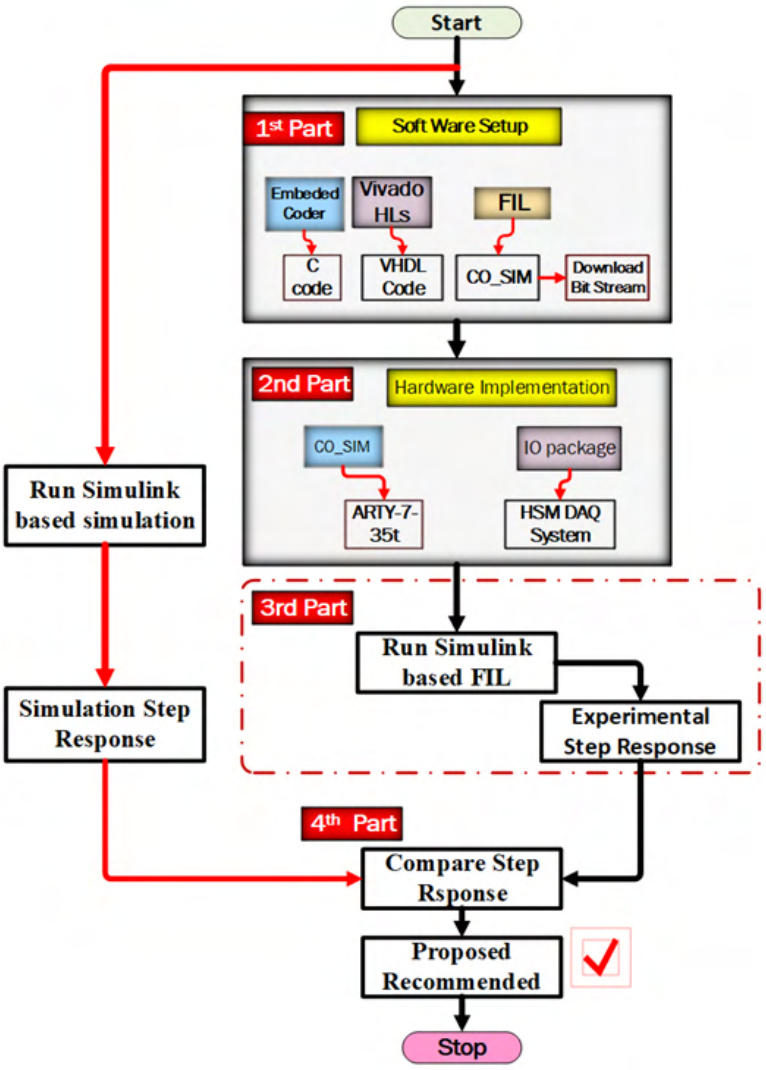

Figure (4). Hardware verification Methodology.

Figure (5) shows the block diagram of the proposed controller. The HSPICS designed to achieve a level of transient state and to examine the steady-state error, by injecting a step signal via HSMDAQ to HSPICS system to acquire the response data of θ. The circuit diagram shows that the Arduino analog write pins A0, A1 was connected with FPGA analog read pins AI and A2, to drive the injecting signals and to the DC motor, and to collect θ by DHES to be measured in scope simulator. Deeply explanation, the HSMDAQ system was utilized to communicate between the DC motor and PC for recording data based scope simulator for visualization and to count encoder's pulses from DHES via Arduino digital input pins D2, D3, where the DHES used to generate a couple of pulses proportionally with θ. The proposed controller employed to control the speed of the motor via Arty 7-35t board through FPGA analog read pin A2, which connected with CMOS RF 520 to drive the motor, thereby connecting and disconnecting the motor to a DC Voltage source depending on the level of the injected signal. Figure (6) presents the hardware setup and block diagram of the HSPICS based FPGA_SoC Arty7-35t. Physically, the FPGA board connected with PC through UART, where the Arduino Uno board was connected through USB com6. There are six devices need to be connected as follows: (1) FPGA_SoC Arty7-35t board connected with PC through UART to download bit stream; (2) Arduino Uno connected with PC through USB com6; (3) it was connected HSMDAQ system FPGA_SoC board via MOSFET RF 520 driver to drive the motor. Power MOSFET RF 520 was employed as on-off switching when a voltage is supplied to the Gate, the circuit between the Source and Drain pins is closes and vice versa; (4) it was added a diode 1N4007 connected in parallel with tested DC motor to prevent damaging motors by back emf; (5) the supplied DC voltage is 12-volt; (6) the encoder of DHES for M1, M2, M3 provides (48, 80, 52) counts per revolution respectively, (for counting both rising and falling edges).

Specifically, motors 1 and 3 include a gearbox so that the DHES generates 1633, 1448 counts per revolution on the output shaft of the gearbox, respectively.

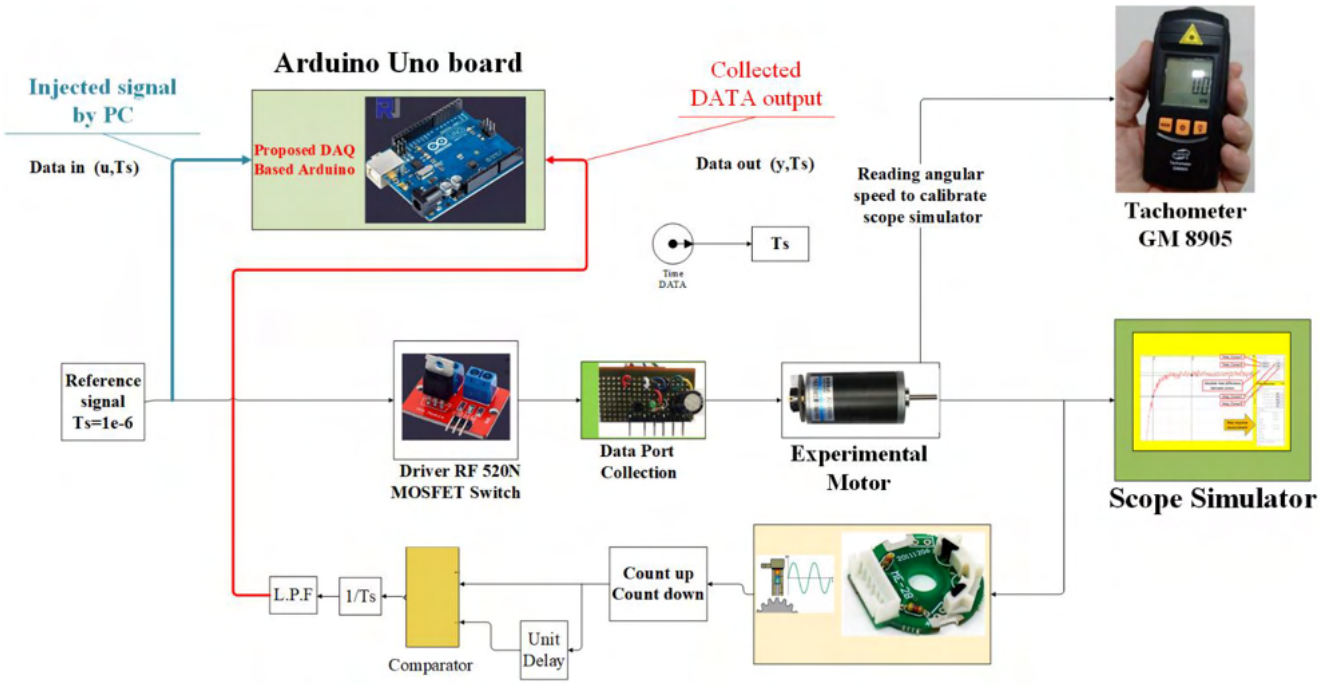

Figure (5). Proposed HSPICS Model.

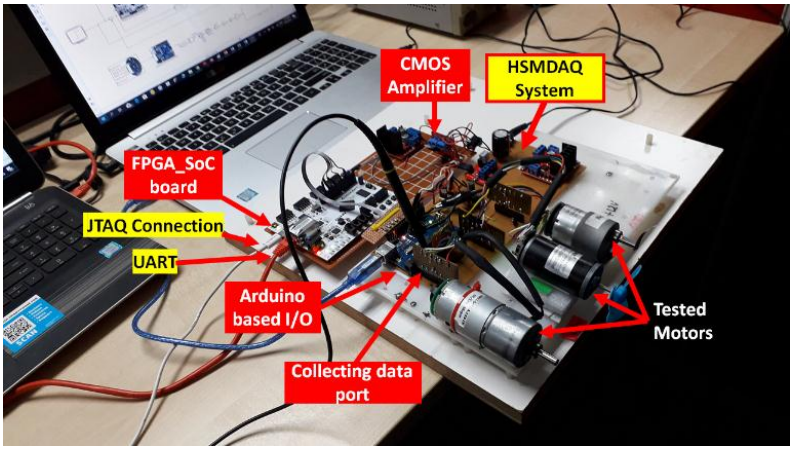

Figure (6). The HSPICS hardware setup.

To fabricate the proposed design, there are three stages were suggested to generate the hardware bit stream code as shown in Figure (7), started from generated C code based Simulink model using Embedded Coder. Next to generate Register Transfer Level (RTL) and VHDL code using Vivado HLs. Lastly, programming FPGA by using the FIL technique to construct CO_SIM for a real time simulation and to download logic bit stream on the Arty7-35t board.

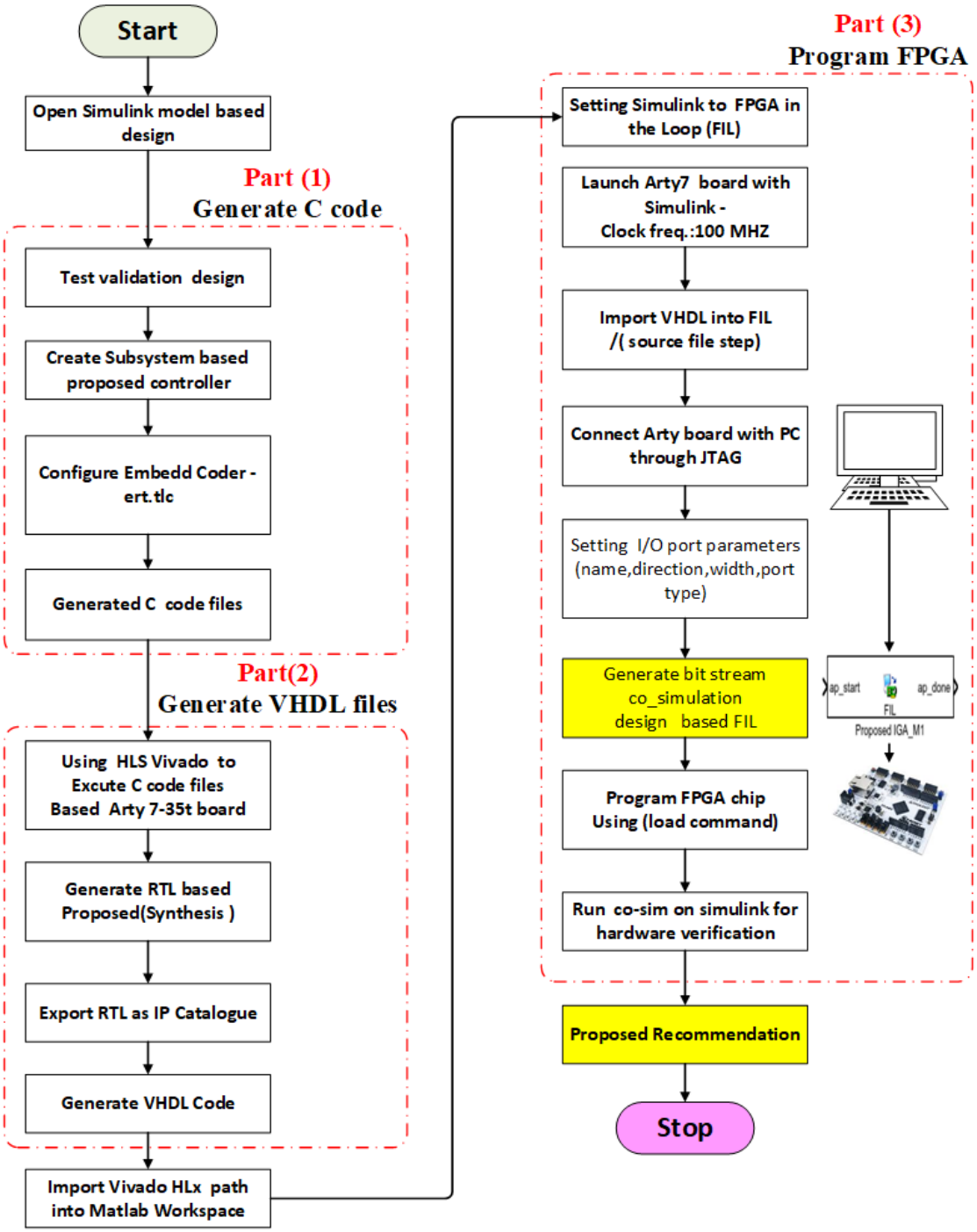

Figure (7). Software setup Methodology.

Figure (8) illustrates the controller Simulink model based proposed IGA to generation C code of hard circuitry. The step reference u(t) used to inject the signal into the motor's armature as an input and the rotational speed of the shaft as the output. Firstly, configuring the Embedded Coder on Simulink to generate C code based targeted board. This facility can be used with subsystems based Simulink model in a real-time denoted a model-based design, which has become widely used for several productive tasks, not only to do simulations test but to provide insight into the dynamic and algorithmic aspects of the system [72] . Based on the aforementioned Figure (4) stage (1), several steps are followed to generate C code based HSPICS as follows:

- Open the controller Simulink model and create subsystem based proposed IGA.
- Run the Simulink based proposed design for checking no error.

- Set simulation time 0.03 with sampling time Ts= 6^{-6}sec.
- Setting Model Configuration Parameters on Code Generation by selected Embedded Coder to generate C code based IGA_PI controller.
- Generate C code files of the created subsystem by Embedded Coder. The C code files named (Propo_IGA_M1_ert_rtw) contained C code and header files.

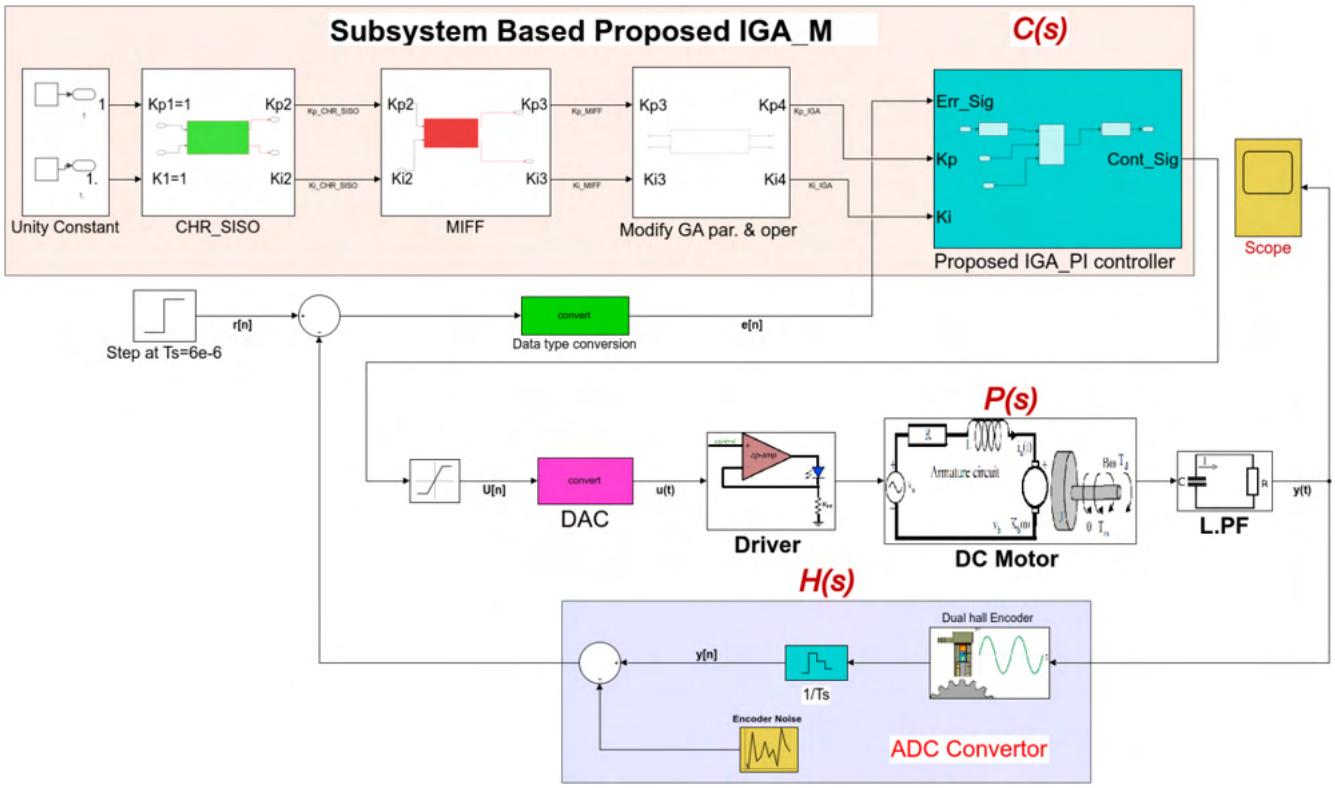

Figure (8). Controller Simulink model based proposed IGA to generate C code.

C. *VHDL code Generation*

The purpose is to presents how to generate VHDL code from C code to configure hardware circuitry on the selected FPGA_SoC board. The suggested method allows to integrating hardware circuitry without needing additional cost design, further providing high accuracy design for implementing complex controller algorithm. By contrast, programming FPGA is a very crucial point in case of FPGA working on a fixed-point algorithm, whereas programming in C language employs for floating-point algorithm [73] . For this reason, it is used Vivado HLs to generate register-transfer-level (RTL) from the generated C code, then importing VHDL files code into the FIL application to configure the proposed hardware on FPGA_SoC. Based on the aforementioned Figure (7) stage (2), there are several steps were suggested to generate VHDL code files as follows:

- Import the c code files into Vivado HLs.
- In order the HLs to generate register transfer level (RTL) code files, the selection board option should be targeted on the Arty 7-35t board.
- Running the HLs application to execute C. code, then to generate RTL
- After generating RTL files design, the HLs will create a new files under the project name, which contains VHDL code of the proposed design. These files automatically named as following :(apc), (setting), (solution_M1).
- From the file (solution _M1), it can be opened to see the file (impl) which contains all details about the proposed design included VHDL and Verilog files code.
- The last step is to build CO_SIM based proposed controller using the FIL application, preparing to download hardware bit stream on Arty 7-35-7-35t board.

D. *Programming FPGA_SoC using FIL Technique*

To deploy the proposed controller on Arty7-35t, it was suggested to using the FIL technique instead of previous deploying methods such as HLxVivado or Xilinx ISE comes in case of easy to use, precise implementation, besides the ability to verify the proposed hardware in a real-time simulation, provides high-quality measurements based scope simulator to verify the proposed controller for both experimental and simulation in terms of step response characteristics. Based on the aforementioned Figure (7) stage (3), there are several steps were proposed to generate to deploy the hardware circuitry on Arty7 35t board as follows:

- Open the same Simulink model as aforementioned in Figure (8) which used previously to generate C-code, then running FIL application from verification.
- From FIL options, lunch Arty-7 -35t, setting FPGA clock frequency 100MHz.
- In Source Files, import VHDL code files into FIL application as follows: (1) IGA_M1_dadd_64ns_cud) ;(2) (IGA_M1_dadddsub_6bkb); (3) IGA_M1_dmul_64nsdEe).
- The process will be started for building CO_SIM based proposed controller that will be appearing at the end process
- At the end process, the FIL application constructs the CO_SIM design and automatically generates I/O port as follows: (1) Input port (ap_clk, ap_ rest, ap_start); Output port (ap_done).
- Afterward, the CO_SIM based Simulink model is ready to generate the bit stream to be integrate proposed HSPICS.

The validation design of CO_SIM can be tested and approved by using the Harness test. The final outline design of the proposed controller based FPGA_SoC illustrated in Figure (9). The outline design contains three input pins, the first pin is ap_clk used to inject the clock cycle through FPGA board, and the third pin ap_start used to inject the signal, where the output signal can be controlled by the output pin ap_idle to drive the motor through CMOS driver switching. Figure (10) shows the interior view of the schematic design hardware circuitry. The selected FPGA_SoC Arty7-35t board contains huge available resources such as DSP48, FF instance, FF registers beside other components. Table 2 shows the low consuming utilization recourses for the proposed hardware controller that were utilized through the fabrication of the proposed controller.

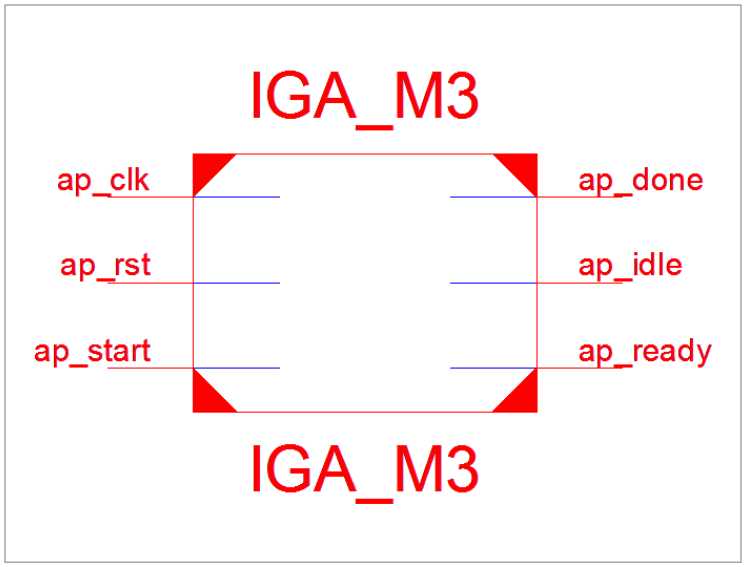

Figure (9). Fabrication pinout diagram of the proposed HSPICS on FPGA_SoC Arty7-35t.

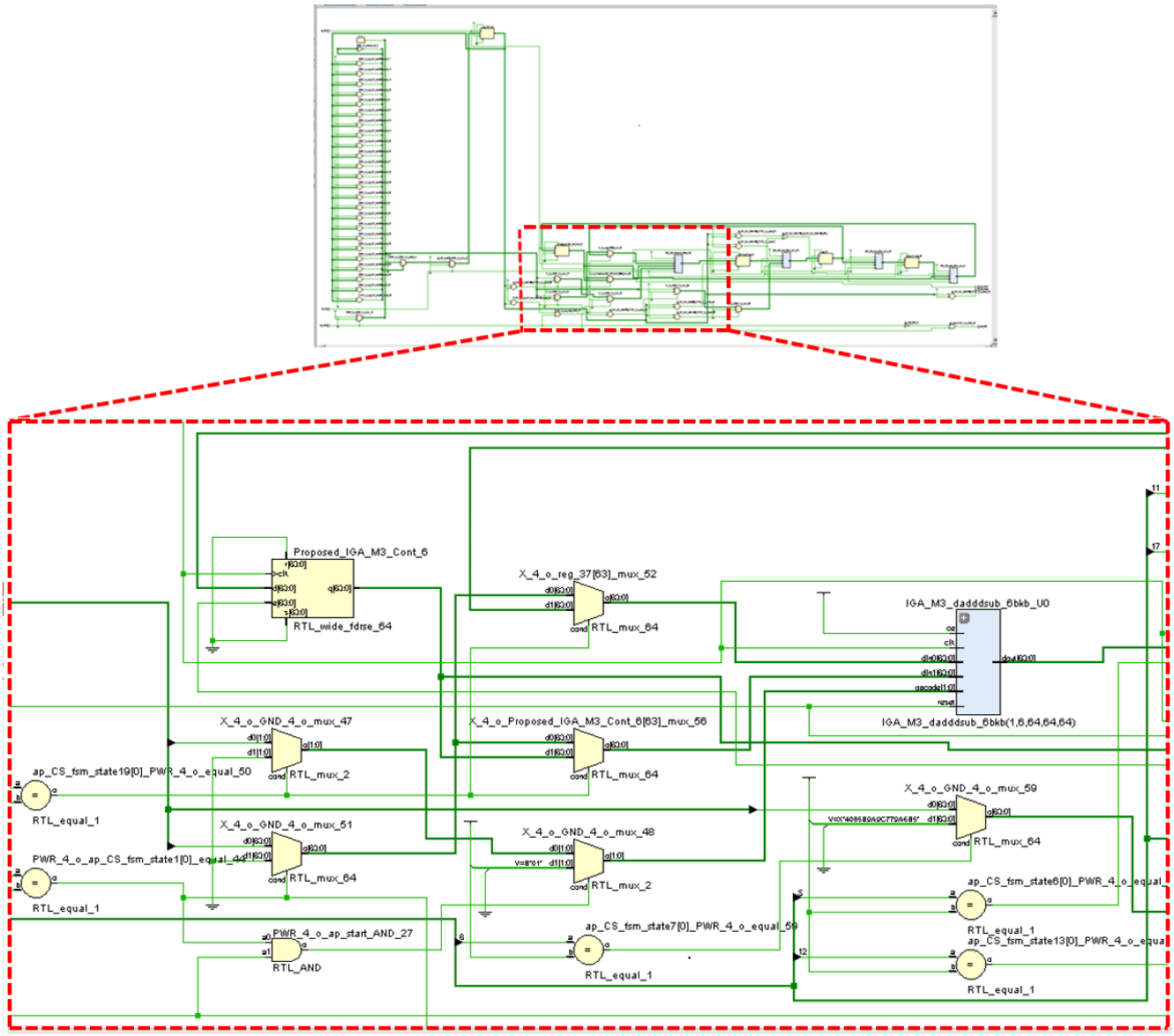

Figure (10). Integration schematic diagram of the proposed HSPICS on FPGA_SoC Arty7-35t.

Table (2). The Consumed Utilization Resources to Integrate HSPICS on FPGA_SoC.

Resources	IGA_M1	IGA_M2	IGA_M3	Available
Latency	23	23	23	XXX
Interval	24	24	24	XXX
DSP48E	28	28	28	90
FF_ Instance	1625	1625	1625	41600
FF_ Register	280	280	280	XXX
LUT_ Instance	2042	2042	2042	20800
LUT_ Multiplexer	215	215	215	XXX
DSP48E%	31	31	31	31
FF%	4	4	4	4
LUT%	10	10	10	10

E. *Hardware Implementation*

Figure (11) demonstrates the methodology to achieve recommended design and to verify the proposed hardware controller. There are five steps need to be followed: (1)connect the FPGA board with HSMDAQ system through data port collection; (2) install ARDUINO block sets on respiratory/MATLAB to run them through PC; (3) running MATLAB as an administrator to manage the hardware circuitry for both HSMDAQ system and FPGA board through CO_SIM design;(4) connect HSMDAQ system with PC through com6 ;(5)running HSMDAQ system

within controller Simulink model to inject signals and to measure step response characteristics using scope simulator. Figure (12) shows the followed steps to verify the hardware using a scope simulator for measuring step response characteristics. There are two PCs were used to deploy the hardware and to run it for measuring td, tr, ts, and SST. The PC2 used to deploy the generated bit stream on Ary7-35t board, where the PC1 used to run the hardware circuitry for both proposed controller and HSMDAQ system. Now, as the signal generator based Simulink model injects signal, the DC motor will spin to collect θ data by collecting data port and to measure the step response characteristics. The values of θ data proportionally with couple pulses generated via DHES through two pins (phases A and B). The Simulink model-based hardware controller using both the IO package and FIL application which employed for three jobs, firstly to control the motor through the switching of the CMOS RF520, secondly to read the encoder output, thirdly to plot the data in real-time using IO package. By running the Simulink model for 1s with sampling time (Ts) 6μs to collect data in- data out for 900 μs, the experimental θ data of the DC motors based proposed controller can be acquired by HSMDAQ system to be imported into MATLAB workspace, where the experimental step response and SST can be plotted by scope simulator.

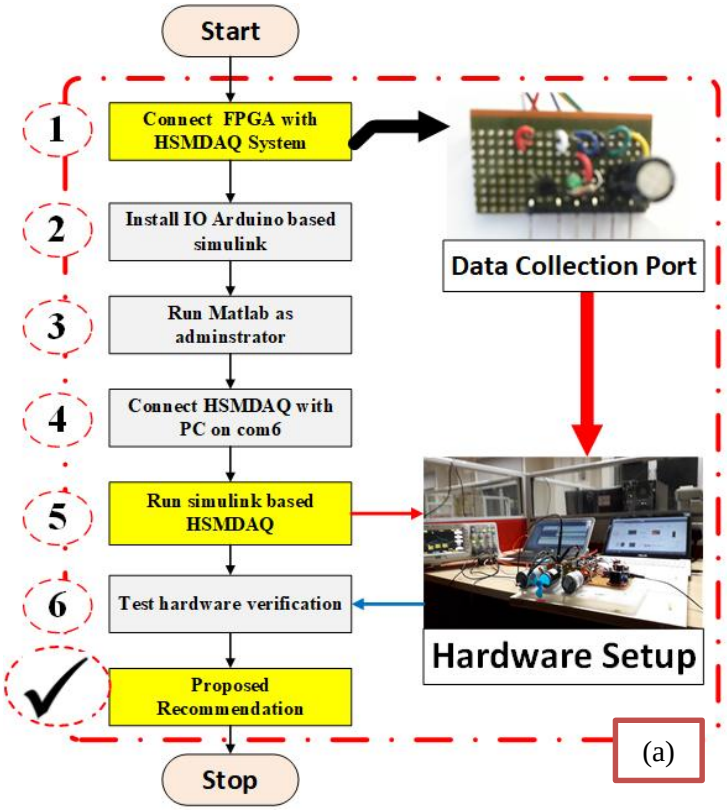

Figure (11). Methodology to achieve recommended design.

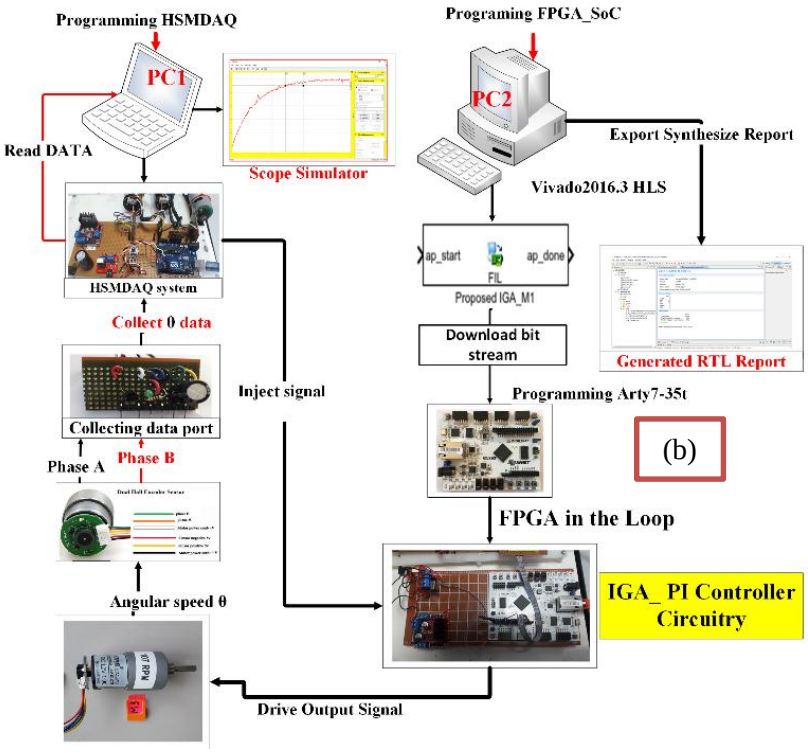

Figure (12). Procedure steps to verify the hardware of the HSPICS controller.

3. Results and Discussion

This section presents the comparison results between simulation and experimental measurements in terms of step response characteristics td, tr, ts, and SST. Firstly, to show the effectiveness simulation results of the proposed IGA algorithm, it was used PSO and classical GA as a benchmark comparison. Secondly, to show the hardware percentage error.

A. *Simulation Results and Benchmark Comparison*

As shown in Figure (13), the comparison step response between the proposed IGA with exiting strategies based classical GA and PSO were presented as a benchmark comparison for M1, M2, M3. The proposed IGA, GA, PSO are conducted to design the PI controller for the DC motor system using the same boundaries by iteration 50, coded by MATLAB running on the same platform. For the PI controller design based PSO and traditional GA, the boundaries of the PI parameters are set to perform the search space as follows: Kp ∈ [0, 400] and Ki ∈ [0, 400]. The design performs the search of 50 trials with different initial solutions to obtain the best solution. After the search process stopped, the PI controller parameters are successfully obtained results by IGA, GA, and PSO. Based on step response results, it can be observed that the proposed IGA based PI controller (IGA_PI) overcomes the classical GA_PI and PSO_PI controller. Table 3 summarized the comparative simulated step response of a PI controller designed by three different algorithms (proposed IGA, conventional GA, and PSO) for the tested motors. The comparative results show the proposed IGA overcomes classical GA and PSO for all tested motors to be measured in the microsecond unit. Also, it can be observed that the PSO produced a better reduction comparing with classical GA for all tested motor. The significant reduction tr accomplished on motor 2 based IGA by 6.77μs for Kp and Ki equal to 78.09 and 679.0 respectively, where the lower reduction tr occurred in motor1 based classical GA by 700.4 μs for Kp 520.19 and Ki 61799.

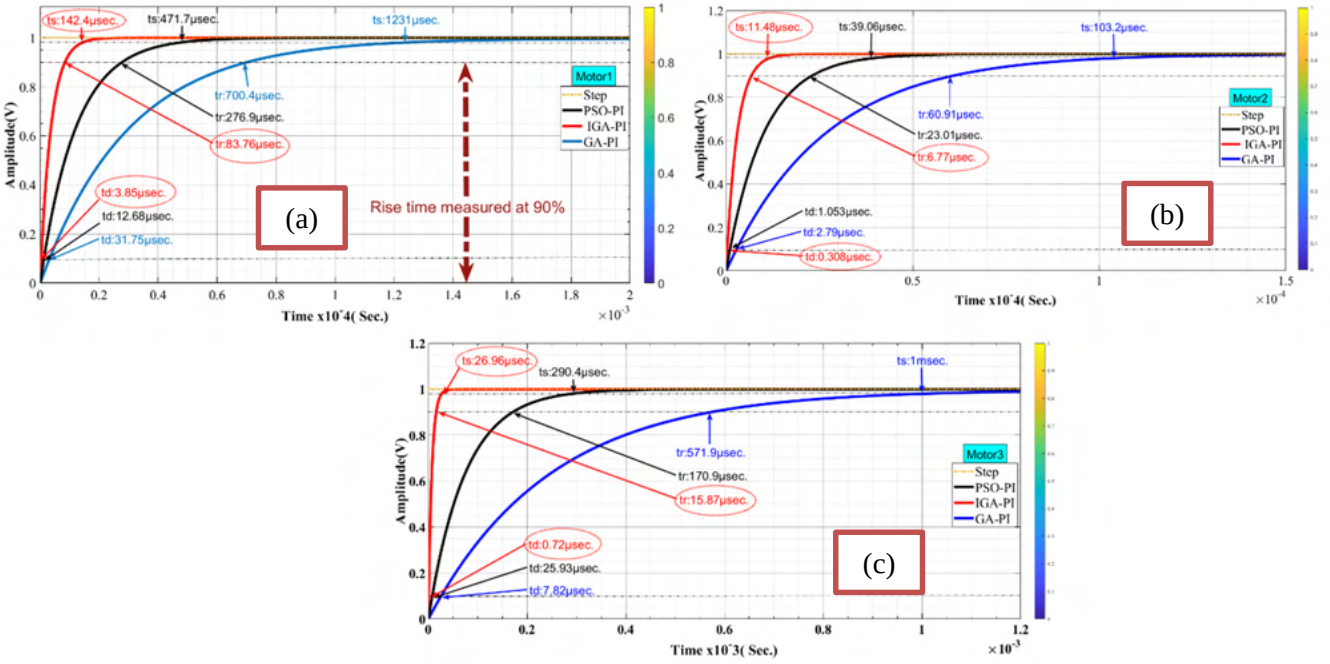

Figure (13). Simulation step response comparison based IGA, GA, and PSO for TF forms, (a) M1, (b) M2, (c)M3.

Table (3). Step Response Coefficients and Proportional Gains Based Proposed IGA, GA and PSO.

M	Algorithm	td(µs)	tr(µs)	ts(µs)	Kp	Ki
1	IGA	3.85	83.76	142.4	4299	701.95
	PSO	12.68	276.9	471.7	1300	414
	GA	31.75	700.4	1227	520.19	617.99
2	IGA	0.3	6.77	11.48	783.09	679.09
	PSO	1.053	23.01	39.06	229.54	363.13
	GA	2.79	60.91	103.2	86.44	136.75
3	IGA	0.72	15.87	26.96	22655	695.2
	PSO	7.821	170.9	290.4	2100	401.89
	GA	25.93	571.9	1000	635	406

B. *Real Time Results*

Figure (14) shows the experimental step response characteristics and speed performance for the tested DC motor (M1, M2, M3) based proposed HSPICS. Based on plotted figures, the proposed controller obtained experimentally much higher reduction step response to be measured in the microsecond unit.

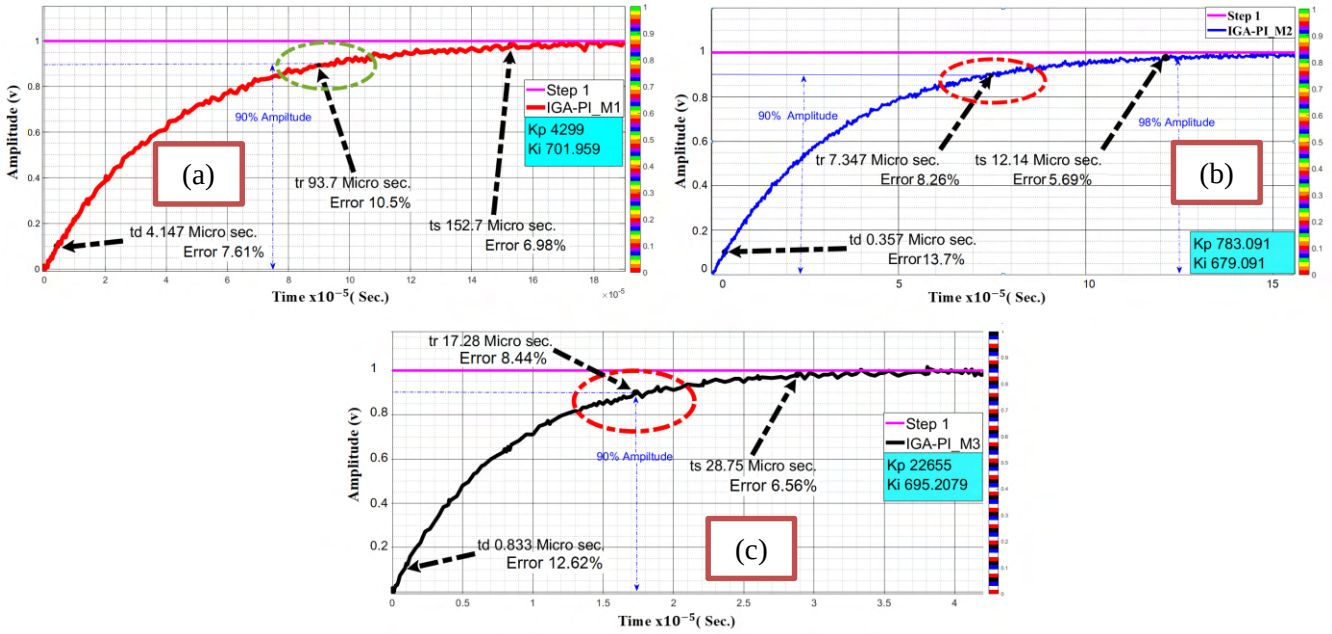

Figure (14). Step response characteristics based HSPICS for the tested DC motors, (a) M1, (b) M2, (c)M3.

As shown in Figure (15) it is observed that the SST for M1, M2, M3 have been minimized by 286.6µs,21.83 µs,41 µs respectively. The highest percentage error appeared in M2 compared with M1 and M3 augmented by 8.67%, where the lowest error achieved in M3 by 3.91%. Significantly, the lowest SST accomplished in motor2 and minimized by 19. 94µs.

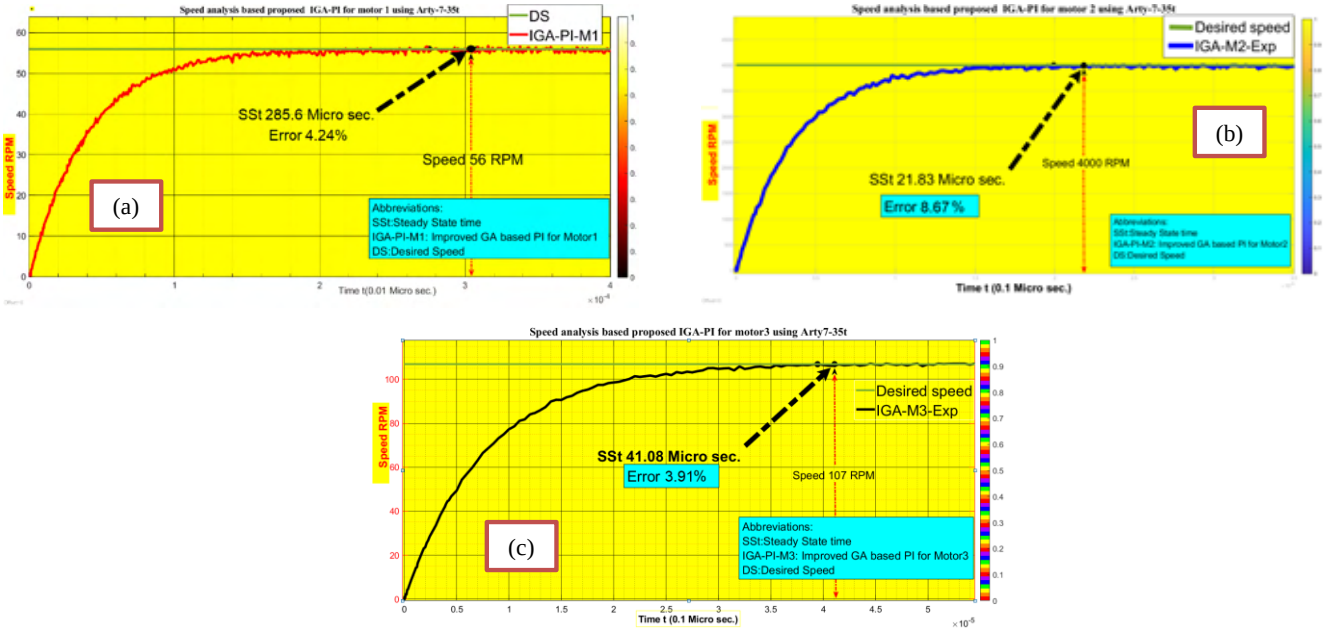

Figure (15). Speed performance analysis based HSPICS for the tested DC motors, (a)M1, (b)M2, (c)M3.

Based on the experimental results discussed in [77], the plant systems (experimental motors without PI controller) are very poor. With the help of the proposed approach, there is a significant improvement in speed performance for both simulation and experiment for all tested DC motors. By contrast, the proposed controller overcomes the classical GA_PI and PSO_PI controller for two reasons. Firstly, it is observed that when modifying the first initialize constraints value in the fitness function, the level of finding better solution increased and the GA produces better solutions than its predecessor, where classical GA relies on random initialization population that causes poor fitness. Secondly, there are many parameters and operators need to be adjusted to get a better solution, while the lousy adjusting causes low performance in classical GA. Simulation results show that the proposed approach

produces a better reduction step response over traditional GA and PSO. Analyzing the performance of the proposed controller comparing with classical GA_PI show that the RSRR for td in motors1,2,3 enhanced by 8,9,35 times, for tr by 8,8,36 times, for ts by 8,8,37 times respectively. Further comparing with PSO_PI, show that the RSRR in terms of td, tr, ts improved by 3,3,10 times for motors 1, 2, and 3, respectively. Based on experimental step response results, it is noticed there is a very low deviation (∆) and magnificent agreement between experimental and simulation for all tested DC motors. The significant experimental results displayed in M2, show that the td, tr, ts minimized by 0.357µs, 7.34µs, and 12.14µs respectively, with magnificent reduction ∆ for td, tr, ts by 48.952ns,0.606µs, 0.691µs respectively. To show the efficiency of the hardware based proposed controller, the percentage error of step response (PESR) for each parameter td, tr, ts can be calculated as given in Eq. (2):

$$\boldsymbol{\Delta t = texperimental - tsimulation} \quad (1)$$

$$\boldsymbol{PESR\% = \left(\frac{\Delta t}{t\ experimental}\right) x100} \quad (2)$$

To calculate the AESR, it was reported the experiment carried out to compare the experimental results with simulation. The AESR could be calculated by dividing the total PESR for each parameter of step response PESR (td), PESR (tr), PESR (ts) by 3 as given in Eq. (3):

$$\boldsymbol{AESR\% = \frac{PESR(td)+PESR(tr)+PESR(ts)}{3}} \quad (3)$$

On the other side, it is tough to obtain the same desired results in simulation and practical. Therefore, there is some error obtained through measuring the experimental step response or SST. But in this approach, there is a significant similarity between experimental and simulation for td, tr, ts with slight difference in AESR between 9.22% to 10.23% for all tested DC motor. The percentage error of SST between simulation and experimental is presented in Table 4. It was noticed whenever increased the reduction in SST especially in values bellow 20µs, the error augmented to above 8%, and this explains why the divergence between simulation and experimental of SST will be increased through running the hardware in microsecond unit for the values bellow than 20µs.

Table (4). Percentage Error Between Simulation and Experimental to Achieve Maximum Speed.

	Proposed Controller	Simulation SST (µs)	Experiment SST(µs)	∆t(µs)	AESR %	RPM
M1	IGA_M1	273.5	285.6	12.11	4.24	56
M2	IGA_M2	19.94	21.83	1.893	8.67	4000
M3	IGA_M3	39.48	41.08	1.607	3.91	107

In the light of these results, it is confidently to say that the proposed hardware controller circuitry based on FPGA_SoC has a stable efficiency, despite using three different DC motor types within same hardware, and this comes from the FPGA's features that can be providing significant advantages over others technologies such as high accuracy performance, parallel processing besides an ability to be used in model-based design technique to reduce the divergence between simulation and experiment.

4. Conclusions

This research mainly constrained improving PI controller for DC motor, to achieve much higher reduction tr and SST to be measured in the microsecond unit. A novel HSPICS based FPGA_SoC was proposed to overcome previous works by taking into consideration all downsides that faced the researchers through designing a PI controller. It was applied multiple techniques to integrate the proposed algorithm on FPGA_SoC, providing magnificent features that could be stated as the following :(1) improve simplicity to integrate a circuitry based algorithm, (2) ease to use, (3) high accuracy design, (4) opening a new path approach to design multi-controller in a single chip. This work involved the simulation and experimental in terms of step response and SST for capturing maximum speed. The main findings are listed in sequence with the objectives as follows:

- To improve GA performance significantly to be used with the PI controller. Where the classical GA has several

drawbacks comes from randomly searching the initial constraint, besides bad selection parameters and operators that cause lousy optimization. It was proposed new MIFF techniques and a new OGA_PI procedure to modify initial constraints and optimize GA parameters and operators. The proposed HSPICS based proposed IGA was applied to a DC motor to show the effectiveness of reduction response time over ordinary GA and PSO. The comparison simulation results show that the proposed IGA overcomes classical GA and PSO for all TF forms of the tested motors. Analysing the performance demonstrates that the RSRR has been enhanced for motors1,2,3 by 8,9,35 times over classical GA, and 3,3,10 over PSO, respectively. By contrast, the proposed controller shows a superior speed performance and displays excellent tuning capability to achieve better step response characteristics. The Proposed IGA_PI controller can be used for wide range applications as well as offering the best tool to improve future controller products.

- To validate the speed performance of the proposed controller design based on step response characterization through simulation and experiment using Arty7-35t board. Experimentally, it was observed that the significant reduction accomplished for all tested motors. The SST for M1, M2, M3 was minimized by 4.24%, 8.67%, 3.91%, respectively, and the deviation of SST between simulation and experimental minimized by12.11µs, 1.8µs, 1.6µs respectively The significant speed performance was achieved in M2 to capture maximum speed 4000 RPM at 21.83µs, where M1 and M3 accomplished maximum speed at 285.6 µs and 41.08 µs respectively.
- To propose a modern method to generate VHDL based Simulink model and to achieve high accuracy hardware design. It was used various applications such as Embedded Coder, HLs Vivado to generate C , VHDL code, and bit stream to integrate the proposed controller, then using the FIL technique to run the design based Simulink model.

The fabrication HSPICS shows very low utilization resources were consumed. It is recommended to implemented a massive PI controller on a single chip that could be employed significantly in field of networking robotic controller, offering high speed performance, low area of design, fidelity, reduce time design further providing the highest accuracy performance to overcomes many previous controller techniques. In future work, the authors plan to find way to apply the proposed HSPICS on Industrial multi robotic arms to increase the productivity and accuracy.

Acknowledgement: Authors would like to thanks to UPM for supporting and funding this research.

References

[1] B. Vaisi, "A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and beyond," *Decis. Anal. J.*, vol. 2, no. October 2021, pp. 1–18, 2022.
[2] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control Strategies for Soft Robotic Manipulators: A Survey," *Soft Robot.*, vol. 5, no. 2, pp. 149–163, 2018.
[3] F. S. Alkhafaji, W. Z. W. Hasan, N. Sulaiman, and M. B. T. Maryam, "Design and Implementation A Novel System For Estimation Precise Transfer Function of DC Motor," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 5, pp. 1118–1125, 2020.
[4] F. S. M. Alkhafaji, W. Zuha Wan Hasan, N. Sulaiman, and M. Mohd Isa, "A Novel PID Robotic for Speed Controller Using Optimization Based Tune Technique," in *Computational Optimization Techniques and Applications employed*, vol. 32, 2021, pp. 1–22.
[5] R. K. Achanta and K. PamulaVinay, "DC Motor Speed Control using PID Controller Tuned by Jaya Optimization Algorithm," *IEEE Int. Conf. Power, Control. Signals Instrum. Eng.*, pp. 983–987, 2017.
[6] Subiyanto, S. Alfa Faridh, and T. W. I. Khadari, "Performance Simulation Of Various Intelligent Techniques For DC Motor Speed Control," *OSR J. Electr. Electron. Eng.*, vol. 13, no. 4, pp. 14–26, 2018.
[7] K. Manjeet, "Literature Review of PID Controller based on Various Soft Computing Techniques," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 6, pp. 1375–1378, 2017.
[8] T. Wang, H. Wang, C. Wang, and H. Hu, "A novel PID controller for BLDCM speed control using dual fuzzy logic systems with HSA optimization," Nature Publishing Group UK, 2022.
[9] N. Maher, G. A. Elsheikh, A. N. Ouda, W. R. Anis, and T. Emara, "Design and Implementation of a Wireless Medical Robot for Communication Within Hazardous Environments," *Wirel. Pers. Commun.*, vol. 122, no. 2, pp. 1391–1412, 2022.
[10] V. K. Tiwari, S. Prakash, and A. Zeeshan, "Speed Control of Induction Motor Fed From Wind Turbine Using Genetic Algorithm," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 3, no. 8, pp. 11457–11465, 2014.

[11] S. Das, A. Chakraborty, J. K. Ray, S. Bhattacharjee, and B. Neogi, "Study on Different Tuning Approach with Incorporation of Simulation Aspect for Z-N (Ziegler-Nichols) Rules," *Int. J. Sci. Res. Publ.*, vol. 2, no. 8, pp. 1–5, 2012.
[12] K. H. Raut and S. Vaishnav, "Performance Analysis of PID Tuning Techniques based on Time Response specification," *Ijireeice.Com*, vol. 2, no. 1, pp. 0–3, 2014.
[13] J. C. Mugisha, B. Munyazikwiye, and H. R. Karimi, "Design of Temperature Control System Using Conventional PID and Intelligent Fuzzy Logic Controller," *iFUZZY 2015 - 2015 Int. Conf. Fuzzy Theory Its Appl. Conf. Dig.*, pp. 50–55, 2016.
[14] T. Hägglund and K. J. Åström, "Revisiting The Ziegler-Nichols Tuning Rules For PI Control," *Asian J. Control*, vol. 6, no. 4, pp. 469–482, 2004.
[15] S. M and L. Moonyong, "IMC Filter Design for PID Controller Tuning of Time Delayed Processes," in *IntechOpen*, 2016, pp. 253–286.
[16] O. Aydogdu and M. Korkmaz, "A Simple Approach to Design of Variable Parameter Nonlinear PID Controller," *Int. Conf. Adv. Inf. Technol. with Work. ICBMG*, vol. 20, pp. 81–85, 2011.
[17] M. M. R. A. Milani, T. Çavdar, and V. F. Aghjehkand, "Particle swarm optimization - Based determination of Ziegler-Nichols parameters for PID controller of brushless DC motors," *INISTA 2012 - Int. Symp. Innov. Intell. Syst. Appl.*, 2012.
[18] T. Slavov and O. Roeva, "Application of Genetic Algorithm to Tuning a PID Controller for Glucose Concentration Control," *WSEAS Trans. Syst.*, vol. 11, no. 7, pp. 223–233, 2012.
[19] G. Soumya, R. D. Darbar, and B. N. Neogi, "Application of Swarm Intelligence Computation Techniques in PID Controller Tuning: A Review," *Commun. Comput. Inf. Sci.*, vol. 292, pp. 194–208, 2012.
[20] S. Inthiyaz, R. Nalli, T. Rakesh, K. Subbarao, S. H. Ahammad, and V. Rajesh, "GA based PID controller: Design and Optimization," *Proc. 6th Int. Conf. Inven. Comput. Technol. ICICT 2021*, pp. 285–289, 2021.
[21] F. Alkhafaji, "Modeling and Control High Speed Robotic Arm for Industrial Applications," in *Global Congress on Electrical Engineering (GC-ElecEng) , Valencia,Spain*, 2021, pp. 1–10.
[22] Q. Tu, H. Li, X. Wang, and C. Chen, "Ant Colony Optimization for the Design of Small-Scale Irrigation Systems," *Water Resour. Manag.*, vol. 29, no. 7, pp. 2323–2339, 2015.
[23] S. M. G. Kumar, R. Jain, N. Anantharaman, V. Dharmalingam, and K. M. M. S. Begum, "Genetic Algorithm Based PID Controller Tuning for a Model Bioreactor," *Indian Chem. Eng.*, vol. 50, no. 3, pp. 214–226, 2008.
[24] M. Sazli Saad, H. Jamaluddin, and Z. Darus Mat, "Implementation of PID Controller Using Diffrential Evolution and Genetic Algorithm," *Int. J. Inov. Comput. ,Information Control*, vol. 8, no. 11, pp. 7761–7779, 2012.
[25] H. Ali and S. Wadhwani, "Intelligent PID Controller Tuning for Higher Order Process System," *Int. J. u- e-Service, Sci. Technol.*, vol. 8, no. 6, pp. 323–330, 2015.
[26] L. Ma, Y. Yao, and M. Wang, "The Optimizing Design of Wheeled Robot Tracking System by PID Control Algorithm Based on BP Neural Network," *Proc. - 2016 Int. Conf. Ind. Informatics - Comput. Technol. Intell. Technol. Ind. Inf. Integr. ICIICII 2016*, pp. 34–39, 2017.
[27] K. O. Jones and W. Hengue, "Limitations of multivariable controller tuning using genetic algorithms," *Int. Conf. Comput. Syst. Technol. - CompSysTech*, 2010.
[28] R. R. Mohd, S. Hazlina, and Z. Hairi, "Multi-Objective Optimization for PID Controller Tuning Usin The Global Ranking Genetic Algorithm," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 1, pp. 269–284, 2012.
[29] B. Hekimoglu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, pp. 38100–38114, 2019.
[30] S. P. Singh, K. Kumar, S. K. S. Verma, J. Singh, and N. Tiwari, "A Review on Control of a Brushless DC Motor Drive," *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng.*, vol. 4, no. 1, pp. 82–97, 2018.
[31] A. Batool, N. ul Ain, A. A. Amin, M. Adnan, and M. H. Shahbaz, "A comparative study of DC servo motor parameter estimation using various techniques," *Automatika*, vol. 63, no. 2, pp. 303–312, 2022.
[32] T. A. Tutunji, "DC Motor Identification using Impulse Response Data," *EUROCON 2005*, vol. 00, pp. 1734–1736, 2006.
[33] F. S. M. Alkhafaji, W. Z. Wan Hasan, M. M. Isa, and N. Sulaiman, "A response time reduction for DC motor controller using SISO technique," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 895–906, 2020.
[34] M. Fruk, G. Vujisić, and T. Špoljarić, "Parameter Identification of Transfer Functions Using MATLAB," *2013 36th Int. Conv. Inf. Commun. Technol. Electron. Microelectron.*, no. January 2017, pp. 571–576, 2013.
[35] H. Pan, J. Yao, and M. Gu, "Motor Parameter Measurement and Testing System Based on Labview," *2nd Int.*

Conf. Comput. Appl. Syst. Model., pp. 1349–1351, 2012.
[36] T. A. Tutunji and A. Saleem, "A methodology for identi fi cation and control of electro-mechanical actuators," *MethodsX*, vol. 2, pp. 219–231, 2015.
[37] D. Ramasubramanian, "Identification and Control of DC Motors," Escola Tècnica Superior d'Enginyeria Industrial de Barcelona, 2016.
[38] W. L. Oberkampf and M. F. Barone, "Measures of agreement between computation and experiment: Validation metrics," *J. Comput. Phys.*, vol. 217, no. 1, pp. 5–36, 2006.
[39] T. Magraner, Á. Montero, S. Quilis, and J. F. Urchueguía, "Comparison Between Simulation and Experimental Results for The Energy Performance of Geocool Geothermal Experimental Plant," *Proc. 11th Int. Conf. Therm. Energy Storage*, no. June, 2009.
[40] K. Monmoto *et al.*, "Dynamic diagnostic and reconfiguration for a new FPGA based green controller," *2015 Int. Conf. Renew. Energy Res. Appl. ICRERA 2015*, pp. 977–982, 2015.
[41] M. L. Ruz, J. Garrido, F. Vazquez, and F. Morilla, "Interactive tuning tool of proportional-integral controllers for first order plus time delay processes," *Symmetry (Basel).*, vol. 10, no. 11, pp. 1–22, 2018.
[42] Z. M. Kassas, "Methodologies for implementing FPGA-based control systems," *IFAC Proc. Vol.*, vol. 44, no. 1 PART 1, pp. 9911–9916, 2011.
[43] V. Gupta, K. Khare, and R. P. Singh, "Efficient FPGA implementation of 2 nd order digital controllers using matlab/simulink," *J. Eng. Appl. Sci.*, vol. 6, no. 8, pp. 94–99, 2011.
[44] A. Thangavelu, M. V. Varghese, and M. V. Vaidyan, "Novel FPGA based controller design platform for DC-DC buck converter using HDL Co-simulator and Xilinx System Generator," *ISIEA 2012 - 2012 IEEE Symp. Ind. Electron. Appl.*, pp. 270–274, 2012.
[45] F. S. M. Alkhafaji, W. Z. W. Hasan, M. M. Isa, and N. Sulaiman, "Robotic Controller : ASIC versus FPGA — A Review," *J. Comput. Theor. Nanosci.*, vol. 15, no. 1, pp. 1–25, 2018.
[46] Z. A. Abbas, N. B. Sulaiman, N. A. M. Yunus, W. Z. Wan Hasan, and M. K. Ahmed, "An FPGA implementation and performance analysis between Radix-2 and Radix-4 of 4096 point FFT," *2018 IEEE 5th Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2018*, pp. 2–5, 2019.
[47] K. Shahbazi and S. B. Ko, "High throughput and area-efficient FPGA implementation of AES for high-traffic applications," *IET Comput. Digit. Tech.*, vol. 14, no. 6, pp. 344–352, 2020.
[48] X. Huang, Z. Guo, M. Song, and X. Zeng, "Accelerating the SM3 hash algorithm with CPU-FPGA Co-Designed architecture," *IET Comput. Digit. Tech.*, vol. 15, no. 6, pp. 427–436, 2021.
[49] E. Ishii, H. Nishi, and K. Ohnishi, "Improvement of performances in bilateral teleoperation by using FPGA," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1876–1884, 2007.
[50] O. Oballe-Peinado, J. Castellanos-Ramos, J. A. Hidalgo, F. Vidal-Verdú, H. Macicior, and E. Ochoteco, "Interface for tactile sensors based on direct connection to a FPGA," *IEEE 2009 Int. Conf. Mechatronics, ICM 2009*, 2009.
[51] C. Trabelsi, S. Meftali, and J.-L. Dekeyser, "Semi-distributed Control for FPGA-based Reconfigurable Systems," *2012 15th Euromicro Conf. Digit. Syst. Des.*, pp. 185–192, 2012.
[52] I. Bravo-Muñoz, A. Gardel-Vicente, and J. L. Lázaro-Galilea, "New applications and architectures based on FPGA/SoC," *Electron.*, vol. 9, no. 11, pp. 1–4, 2020.
[53] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informatics*, vol. 7, no. 2, pp. 224–243, 2011.
[54] C. Economakos, M. Skarpetis, and G. Economakos, "Program-based and Model-based PLC Design Environment for Multicore FPGA Architectures," pp. 726–733, 2014.
[55] A. Ponsich, C. Azzaro-Pantel, S. Domenech, and L. Pibouleau, "Constraint handling strategies in Genetic Algorithms application to optimal batch plant design," *Chem. Eng. Process. Process Intensif.*, vol. 47, no. 3, pp. 420–434, 2008.
[56] B. Singh, S. Prakash, A. S. Pandey, and S. K. Sinha, "Intelligent PI Controller for Speed Control of D . C . Motor," *Control*, vol. 2, no. 1, pp. 87–100, 2010.
[57] S. M. Girirajkumar, D. Jayara, and KishanAnoop.R., "PSO based Tuning of a PID Controller for a High Performance Drilling Machine," *Int. J. Comput. Appl.*, vol. 1, no. 19, pp. 12–18, 2010.
[58] M. G. Dozein, A. Gholami, and M. Kalantar, "Speed Control of DC Motor Using Different Optimization Techniques Based PID Controller," *IEEE Int. Conf. Eng. Technol.*, vol. 2, no. 7, pp. 6488–6494, 2012.
[59] A. H. Fathi, H. Khaloozadeh, M. A. Nekoui, and ShisheieShisheie, "Using PSO and GA for Optimization of PID Parameters," *Int. J. Intell. Inf. Process.*, vol. 3, no. 1, pp. 71–78, 2012.

[60] S. Karthikeyan, P. Rameshbabu, and J. Robi, "Design of intelligent PID controller based on Particle Swarm Optimization in FPGA," *Int. J. Power Control Signal Comput.*, vol. 4, no. 2, pp. 66–70, 2012.
[61] N. L. S. Hashim, A. Yahya, T. Andromeda, M. R. A. Kadir, N. Mahmud, and S. Samion, "Simulation of PSO-PI Controller of DC Motor in Micro-EDM System for Biomedical Application," *Procedia Eng.*, vol. 41, pp. 805–811, 2012.
[62] K. Latha, V. Rajinikanth, and P. M. Surekha, "PSO-Based PID Controller Design for a Class of Stable and Unstable Systems," *ISRN Artif. Intell.*, vol. 2013, pp. 1–11, 2013.
[63] F. Mahar, S. M. A. Shah, and S. Khan, "PSO and GA Based Fixed Order Controller Design:Comparison and and Performance Analysis," *Int. Conf. Innov. Eng. Technol.*, pp. 16–21, 2013.
[64] M. Kishnani, S. Pareek, and R. Gupta, "Optimal Tuning of PID Controller Using Genetic Algorithm and Swarm Techniques," *Int. J. Electron. Electr. Eng.*, vol. 7, no. 2, pp. 189–194, 2014.
[65] A. T. El-Deen, A. A. Hakim Mahmoud, and A. R. El-Sawi, "Optimal PID tuning for DC motor speed controller based on genetic algorithm," *Int. Rev. Autom. Control*, vol. 8, no. 1, pp. 80–85, 2015.
[66] A. M, L. S, and R. T. H, "Speed Control for Separately Excited DC Motor With PID Controller, GA and PSO," *Int. J. Recent Sci. Res.*, vol. 7, no. 7, pp. 12673–12678, 2016.
[67] S. K. Suman and V. K. Giri, "Speed control of DC motor using optimization techniques based PID Controller," *Proc. 2nd IEEE Int. Conf. Eng. Technol. ICETECH 2016*, pp. 581–587, 2016.
[68] B. Khalfa and C. Abdelfateh, "Optimal tuning of fractional order Pλ D μA controller using Particle Swarm Optimization algorithm," in *IFAC-PapersOnLine*, 2017, vol. 50, no. 1, pp. 8084–8089.
[69] F. S. M. Al Khafaji, W. Z. Wan Hasan, M. M. Isa, and N. Sulaiman, "A HSMDAQ System for EstimatingTransfer Function of a DC motor," *Asia Pacific Conf. Postgrad. Res. Microelectron. Electron.*, vol. 2019-Novem, pp. 25–28, 2019.
[70] F. S. M. Alkhafaji, W. Z. W. Hasan, M. M. Isa, and N. Sulaiman, "A Modified GA based PI controller for DC Motor Performance," in *Proc. of the 6th IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2019, no. August, pp. 27–29.
[71] F. S. M. Alkhafaji, W. Z. Wan Hasan, M. M. Isa, and N. Sulaiman, "Proposed a Novel Method for Optimization DC Motor Controller," in *IEEE 5th International Conference on Smart Instrumentation, Measurement and Application, ICSIMA*, 2019, no. November, pp. 1–4.
[72] G. Sandmann and R. Thompson, "Development of AUTOSAR software components within model-based design," *SAE Tech. Pap.*, no. April 2008, 2008.
[73] D. Bagni and D. Mackay, "Floating-Point PID Controller Design with Vivado HLS and System Generator for DSP," 2013.