Defence Science Journal, Vol. 74, No. 1, January 2024, pp. 91-99, DOI : 10.14429/dsj.74.18983

Evolvable Hardware-Based Optimal Position Control of Quadcopter

Harish Bhat N., Shreesha Chokkadi* and Satish Shenoy B.

Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal - 576 104, India

**E-mail: shreesha.c@manipal.edu*

ABSTRACT

Trading off performance metrics in control design for position tracking is unavoidable. This has severe consequences in mission-critical systems such as quadcopter applications. The controller area and propulsion energy are conflicting design parameters, whereas the reliability and tracking speed are related metrics to be optimized. In this research, a switching-based position controller was co-simulated with the quadcopter model. Performance analysis of the Field Programmable Gate Array (FPGA)-based controller validates a better scheme for tracking speed, propulsion energy, and reliability optimization under similar error performance. To improve the computation power and controller area, the dynamic partial reconfiguration(DPR) approach has been adapted and implemented on FPGA using the Vivado Integrated Development Environment (IDE), where a ranking-based approach brings into action either proportional derivative, sliding mode, or model predictive controllers for each dimension of position tracking. It is verified by analyzing the cumulative tracking speed, reliability, controller area, and propulsion energy metrics that the proposed controller can optimize all these metrics within three successive iterations of tracking either in the same direction or in any combination of directions. Concerning the implementation results of the controller with the switching-based controller, there is 6 % computation power and 30 % resource savings due to DPR.

Keywords: Evolvable hardware; Quadcopter position control; Dynamic partial reconfiguration; Optimal control

NOMENCLATURE

FPGA	: Field Programmable Gate Array
IDE	: Integrated Development Environment
DPR	: Dynamic partial reconfiguration
SMC	: Sliding mode controller
MPC	: Model Predictive controller
PID	: Proportional integral derivative
RTR	: Run Time Reconfiguration
PLA	: Programmable Logic Array
cs, cr, ca, cp	: Cumulative speed, reliability, area, power cnp, cns, cnm: Number of PD/SMC/MPC
HDL	: Hardware Description Language
RM	: Reconfigurable module
LUT/FF	: Look Up Table/Flip Flop
DSP	: Digital Signal Processor

1. INTRODUCTION

Quadcopters, a type of multirotor unmanned aerial vehicle with 4 rotors to control their motion, are powered by a battery that limits the endurance of the quadcopter because they are bulky. With computation limits on board, these are controlled by simple, fixed controllers such as PD. These controllers have larger transients with higher energy demands. Hence,to improve endurance, complex control algorithms with better transients and tracking speeds are options, but these need a larger chip area, which is underutilised for normal tracking needs with reliability issues as well. These conflicting requirements limit quadcopter applications, leading to severe challenges in terms of mission success probability. Hence, there is a need to devise a control approach that optimizes all the desired metrics over longer tracking periods.

FPGA-based PID controllers have been used for second-order systems such as DC motors[1-2]. Performance analysis showed that it has the best controller area, with performance traded off. A PI controller with variable gains for improved tracking, obtained using incremental fuzzy logic, was implemented on FPGA[3]. Moderate improvement was observed when BLDC motor angles were controlled compared to the linear PI controller.

SMC concepts with a control design for distance and time convergence were highlighted, and super twisting control was also explained. PD/PID-based SMCs realized using MATLAB/FPGA were analyzed. PD-based SMC has a better transient response, whereas PID-based SMC has better steady-state performance[5]. FPGA implementation improves timing and overcomes chattering due to its speed. A fractional-order ASMC is designed to control chaos in fractional-order induction motors[6]. The controller is also tested for realization using the Xilinx system generator. The simulation results validate the controller's operation. This controller is complex, with higher chances of faults.

MPC concepts such as prediction/control horizons, the use of models for predicting input, states with input and output disturbances, and their mitigation are highlighted[7]. An explicit MPC with reduced computations using critical regions and searching for outputs in those regions has been designed/

Received : 06 April 2023, Revised : 11 September 2023
Accepted : 06 October 2023, Online published : 12 January 2024

simulated using Simulink, and a realistic CarSim model was developed[8]. It is observed that such a controller can work very fast. FPGA-based implementation further improves speed. FPGA-based MPC was used for current control in a single-phase direct matrix converter[9]. Complex realization results in underutilization for normal tracking needs. A PC with an AirSim simulator for a 3D environment model and control algorithm with landing site detection was implemented on an Arty Z7 FPGA. During testing using hardware in Loop (HiL) simulation, low latency and a 100 % success rate of landing site detection were observed[10].

Hand-coded VHDL implementation requires a longer development time for complex controllers, and a MATLAB-based FPGA is the best option in such cases. For efficient high-level code, suitable data width, pipelining, variable assignment/reuse, and looping optimizations can be considered[11].

The FPGA is a balanced option for design space trade-offs such as flight time, computation time, cost of integration, and fabrication[12]. With need-based control structures using a switching approach, there is a possibility for optimization[13] when controlling robots with Linear/adaptive/FLT/force controllers activated based on tracking error. For passing one control output as the control variable, all controllers activated result in a large computation. Genetic algorithms can be used to evolve PLA-based hardware for common arithmetic and logical operations[14]. Although the architecture is optimized, the large time taken limits the approach in highly complex systems such as robotics.

Phenox, an autonomous robot that estimates position, controls, and follows Linux-based commands and was developed using Zynq 7000 APSoC, was analyzed for power consumption, endurance, and resource utilization[15]. Parallel computations of the tasks ensuring robust operation are possible with FPGA-based quadcopter control[16]. The designed controller with simulations and synthesis is observed to have minimum power and resources.

The FPGA-based MPSoC is the best way to optimally implement AI-based drone control[18]. The reconfiguration concept used for simulation and hardware-based testing is the best alternative to save power. The use of FPGAs, which are reliable and easily developed due to hardware-software co-design, for robotic control design is the reason for their widespread use, even in space missions[19]. The use of DPR is also mentioned to be effective in resource sharing and energy savings[20]. Dynamic reconfiguration is a scheme that can ensure the reliability, optimality, and safety of critical systems such as aerial vehicles[21]. An automated DART tool for various steps in DPR-based system realization can save considerable manual effort and design time[22]. Use, debugging, and large edit-compile-debug cycles are stated as the challenges in RTR-based designs, along with the lack of better abstraction models and support[23]. Chapter 7 of the book on Partial reconfiguration design has summarised partial reconfiguration design steps[24].

Systematic design steps have to be followed for making partial reconfiguration, an easy process, although at the cost of flexibility. For power optimization, two concepts, DPR and voltage scaling, can be considered. Voltage scaling during run time is highly challenging and may result in reliability issues[25]. A summary of the PR process, its implementation steps, and advantages leading to process speedup, savings in board area, and future improvements are discussed[26]. Challenges in PR-based designs, including the architecture, method, management, and hardware adaptability to the need, are addressed in the user guide on partial reconfiguration by Xilinx[27].

Rank assignments for multi objective optimisation problems for candidate solutions is a scheme for multidimensional optimisation. The scheme has the potential to balance tracking concerning diverse performance metrics quickly[28]. To the best of the author's knowledge, as has been reviewed in most of the literature, a controller that meets design requirements with the best possible values for each performance metric quickly is of high priority in mission-

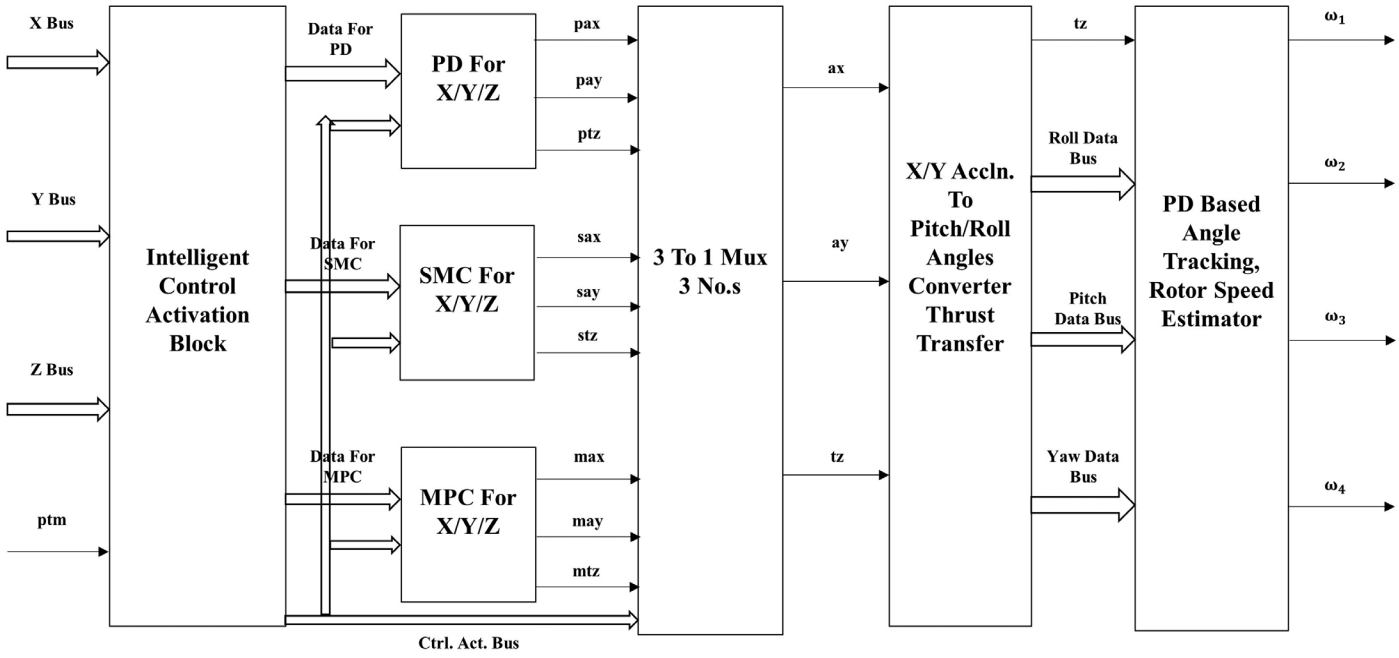

Figure 1. Switching based quadcopter position control system.

critical systems such as quadcopters, which are lacking at this stage. The FPGA, with its DPR-based evolvable hardware scheme, has the potential to address all these challenges.

The rest of the manuscript is outlined as follows. In section II, the switching-based quadcopter position controller is presented, section III covers the expected performance and ranking, section IV discusses the algorithm for control activation, sections V and VI elaborate on the FPGA implementation of switching/DPR-based control using Vivado, section VII presents the co-simulation results and analysis, the implementation results and analysis are discussed in section VIII, and finally, the conclusions and future scope are presented in section IX.

2. SWITCHING-BASED QUADCOPTER POSITION CONTROLLER

As shown in Fig. 1, each of X/Y/Z can be tracked using PD[1], SMC[4], or MPC[7] controllers activated by an intelligent control activation block for the direction of tracking. It activates the controller for optimized performance only if the prioritized tracking mode (ptm) is made inactive (ptm = 0) externally by the user. If ptm is set active by the user, the value set on this signal externally decides the controller to be activated. If ptm=1, the controller area is to be optimized using PD; if ptm is 2, moderate performance/controller area could be achieved with SMC; and if ptm=3, best performance with trade-off in controller area could be achieved with MPC. Other inputs to this block are the X/Y/Z buses with the desired position, the position fed back from the quadcopter, and the expected settling time as individual signals. From these signals, the activation block computes error/delayed error signals. Along with the user-specified expected settling time, these are passed to the individual controllers on the controller's data buses. The control activation bus has three signals cnx, cny, and cnz. A value of 1 was chosen for PD, 2 for SMC, and 3 for MPC. The same signals are passed to 3 muxes to choose accelerations ax/ay from PD/SMC/MPC (pax/pay, sax/say, max/may, respectively), which are used by the angle generator block to produce pitch/roll angles. cnz passes thrust outputs from Z controllers to the rotor speed estimator block. The yaw angle is specified externally, and the desired pitch, roll angles, and angles fed back from the quadcopter, along with delayed angle errors, are passed on corresponding buses to the PD angle controller. These controllers compute pitch/roll/yaw moments, which, along with the thrust signal, are mixed by the rotor speed estimator to obtain the desired rotor speeds that enable attaining the quadcopter's desired positions.

3. EXPECTED PERFORMANCE AND RANKING

Detailed system design and performance analysis, both algorithmically using Simulink and FPGA-based co-simulation using a system generator with the Quad model developed using Simulink, were carried out in our previous works. A rough estimation of the controller area, reliability, speed of tracking, and propulsion energy metrics was estimated. It was noticed that PD has the best controller area and reliability; however, performance in terms of the other two metrics is compromised. Hence, a rank of 1 (best) is assigned for controller area and reliability and 3 (least) for speed and power metrics (PD has a fixed tracking speed). Although the SMC momentarily has a large number of resources utilised, once the control input is computed, it will retain it for half the tracking period, and in the other half, it again computes the control input, which is retained until the end of the tracking period. Hence, it has a moderate rank of 2 for the controller area and reliability metrics. Similarly, due to limited speed and propulsion energy optimization, these metrics are also assigned rank 2. The MPC-based controller with continuous prediction and correction uses comparable resources but continuously. Hence, controller area and reliability metrics are assigned with the lowest rank of 3, whereas with a smooth trajectory for control inputs, propulsion energy, and tracking speed optimization achievable to a maximum level, a rank of 1 for these metrics has been assigned. Table 1 summarises these ranking schemes[28].

4. CONTROL ACTIVATION ALGORITHM

There are two modes of the algorithm for control activation, namely, prioritised and optimized tracking modes, which are explained below.

4.1 Prioritised Tracking Mode

Prioritised Tracking Mode (ptm), there are some instances where it is required to momentarily meet metric targets. Ptm input activated by the user externally enables this, as shown in Fig. 1. A value of either 1, 3, or 2 can be set externally for activating PD/MPC or SMC.

4.2 Optimal Tracking Mode

This mode is entered if the ptm input is zero. It ensures uniform cumulative metrics within three iterations, resulting in balanced tracking. With each controller activated either in X/Y or Z direction, results in instantaneous ranks for speed, reliability, area, and power as per the ranking scheme Table 1. This rank persists until a steady state of tracking is reached. Once reaching, another interval will start whenever a new reference is assigned. Thus, changing the controller results in new instantaneous ranks for the metrics. The controller may also change during the present tracking period due to new references being given for tracking in other directions. Hence, cumulative ranks for each of the metrics resulting in the

Table 1. Ranking scheme

Controllers for X/Y/Z	Rank for Speed	Rank for Reliability	Rank for Area	Rank for Energy
PD	3	1	1	3
SMC	2	2	2	2
MPC	1	3	3	1

addition of successive instantaneous ranks for the total position control system as a whole are computed as in Eqn. (1).

$$\begin{aligned} cs&=(3\cdot cnp)+(2\cdot cns)+cnm \\ cr&=cnp+(2\cdot cns)+(3\cdot cnm) \\ ca&=cnp+(2\cdot cns)+(3\cdot cnm) \\ cp&=(3\cdot cnp)+(2\cdot cns)+cnm \end{aligned} \quad (1)$$

cnp, cns, and cnm are the total number of PD/SMC/MPC controllers activated in various tracking intervals. cs, cr, ca, and cp are the cumulative speed, reliability, area, and power metrics, respectively. For balanced optimal tracking, we have to have the equal minimum possible cumulative metrics. The best solution for the above optimisation problem is equal values for cnp, cns, and cnm. Hence, the control activation algorithm should keep cnp, cns, and cnm equal in successive iterations of tracking. Hence, it is possible to attain balanced tracking within three iterations. Ensuring equal values of cnp/cns/cnm does not require any generations of computation, and the optimization algorithm is without any delay but provides an advanced prediction possibility as well. Flowcharts in Fig. 2 summarise the algorithms of control activation.

2. FPGA-BASED IMPLEMENTATION OF SWITCHING-BASED CONTROL

In the conventional method for balanced position tracking, all the controllers are implemented; however, only the relevant controller is turned on. Many of the resources used in control realization are wasted, along with computation power. The block schematic and the operational principle of this control strategy are discussed in section 2 along with Fig. 1. In this section, the summary of FPGA implementation using Vivado[17] is highlighted.

Verilog modules generated for the fixed-point-based Simulink model need to be added to the Vivado project along with the constraint file (with timing constraints), and a suitable FPGA Zynqultra scale+ device with sufficient resources can be specified. Synthesis, implementation, and bit stream generation can be taken up successively. If there are no timing or power violations and less than 100 % utilisation percentage, FPGA implementation of the switching controller is successful. These steps are summarised in the flowchart in Fig. 3.

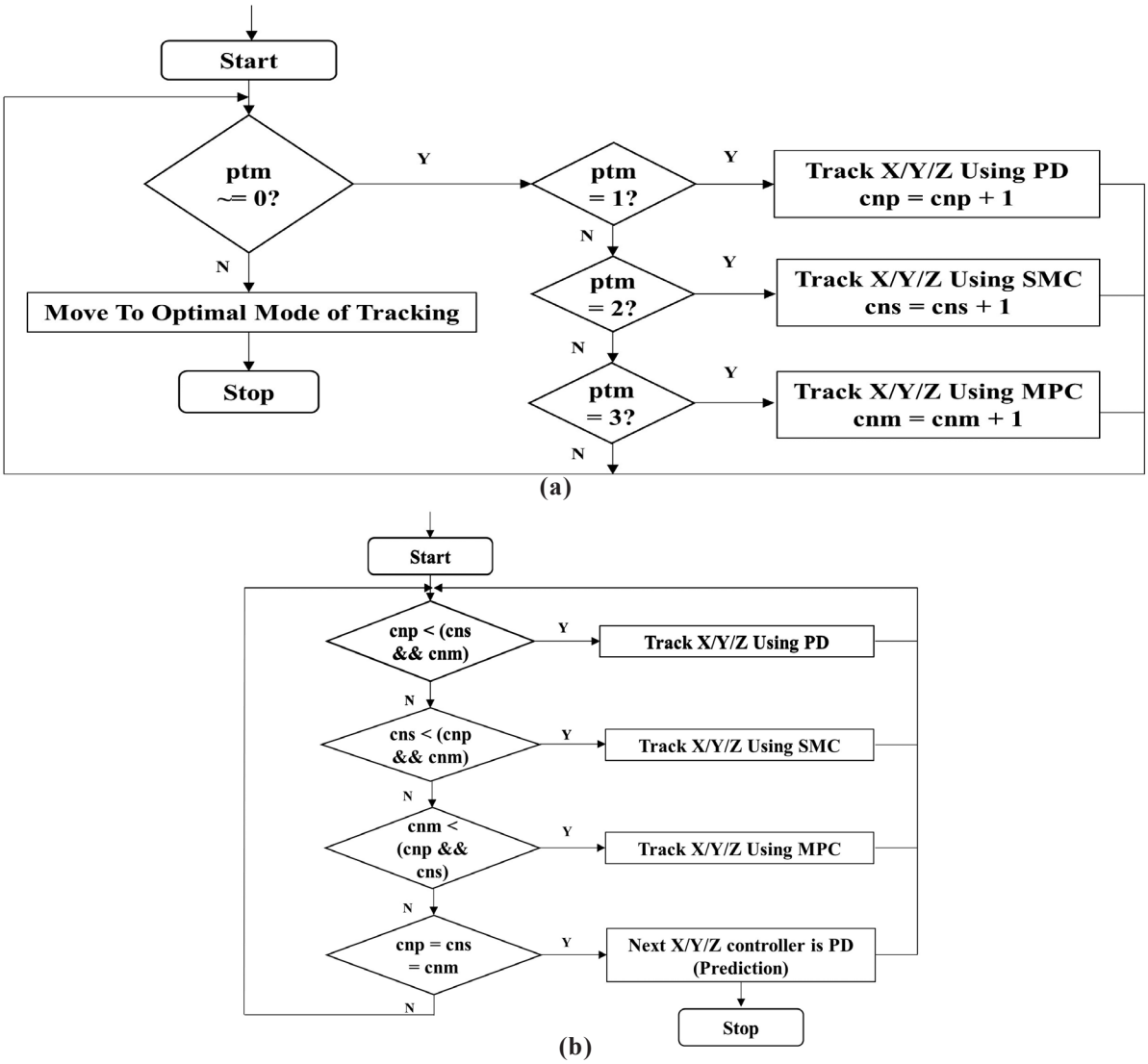

Figure 2. (a) Flowchart for prioritised tracking mode (ptm); and (b) Flowchart for optimised tracking mode.

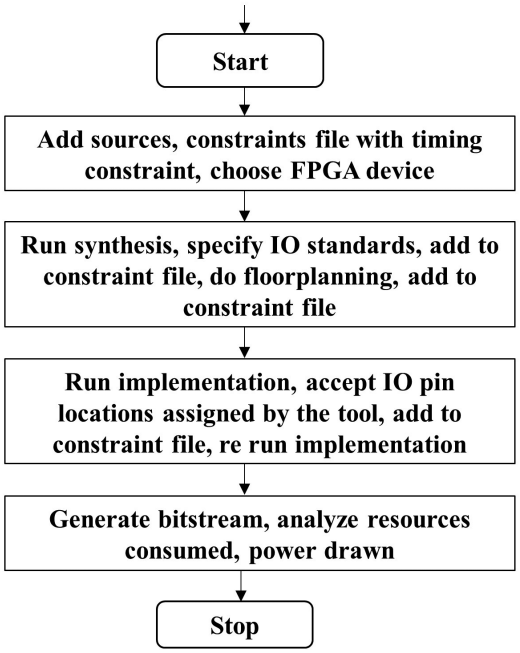

Figure 3. FPGA based implementation of switching based controller

6. DYNAMIC PARTIAL RECONFIGURATION (DPR) FOR EVOLVABLE HARDWARE-BASED POSITION CONTROL

In the switching-based position control system, all the controllers (PD/SMC/MPC) are implemented, but only one of them is turned on according to the algorithm explained in section IV. Most of the metrics can be optimized at the cost of controller area and computation power. Present-day FPGA-based MPSoCs have a concept known as dynamic partial reconfiguration (DPR), where the control architecture evolves itself based on the state machine (here, the intelligent control activation block of section II) developed by the user or sequence set in the processing system. This approach implements only the controller required at the instant, resulting in time sharing of the resources with savings in computational power as well, resulting in the same performance as the switching controller.

With current FPGAs and peripherals, controllers can be reconfigured within milliseconds, resulting in hardware-based subroutines. Vivado v2018.3 supports the DPR approach, where the project created only the controller module with maximum resources to be added, which is considered a reconfigurable module. Static modules are also added, as was done in switching-based controller implementation. Controllers that are going to replace the earlier added controllers are added as children of the parent reconfiguration runs that are going to time-share the partitions on the FPGA device. Once these settings are completed, normal synthesis for the static modules and out-of-context synthesis for the reconfigurable modules, followed by implementation and generation of complete (static + reconfigurable modules) bitstreams and partial bitstreams for each of the reconfigurable modules, are generated. At this stage, further steps of flashing these bitstreams on FPGA for configuration/reconfiguration should be taken. The flowchart in Fig. 4 explains additional steps to be followed for DPR-based implementation[27].

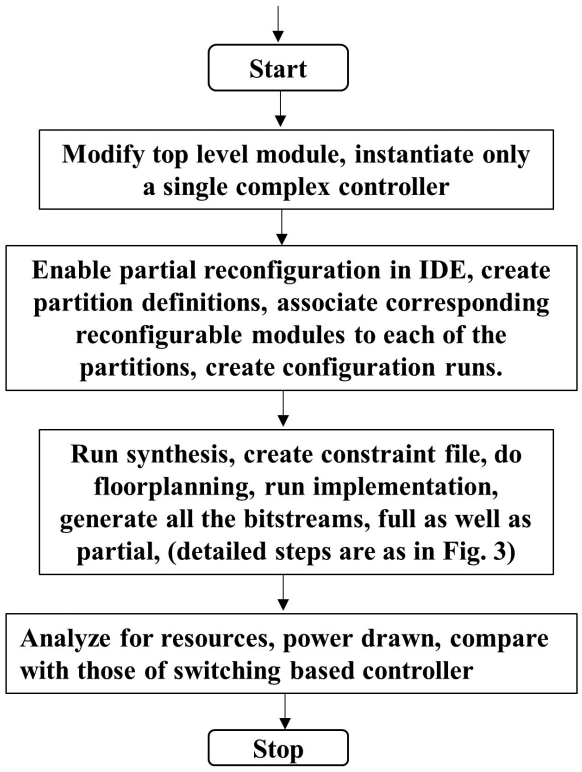

Figure 4. Implementation of DPR based controller on FPGA.

7. COMSIMULATION RESULTS AND ANALYSIS

Performance analysis of the evolvable hardware-based position control system is carried out in 4 stages:

- Co-simulation of the FPGA-based controller model with the MATLAB level 2 s-function-based quadcopter model (For analysis of tracking speed and propulsion energy performance metrics, it should be operated in prioritized tracking mode with inputs as in Table 2.
- Implementation of a switching-based controller for area and reliability metrics analysis on FPGA using Vivado in stage 2.
- Implementation of the DPR-based position controller using Vivado in stage 3.
- The controller is co-simulated in optimal tracking mode along with the quadcopter model, and the cumulative ranks are monitored for successive iterations in stage 4.

Table 2. Input specifications for the controller's speed/energy metric analysis

ptm	Controller activated	Source/destination	References rate	Desired settling time
1	PD	(0,0,0) to (5,5,5) and back	1.2 m/s, Ramp	4 sec
2	SMC	(0,0,0) to (5,5,5) and back	Step	4 sec
3	MPC	(0,0,0) to (5,5,5) and back	Step	4 sec

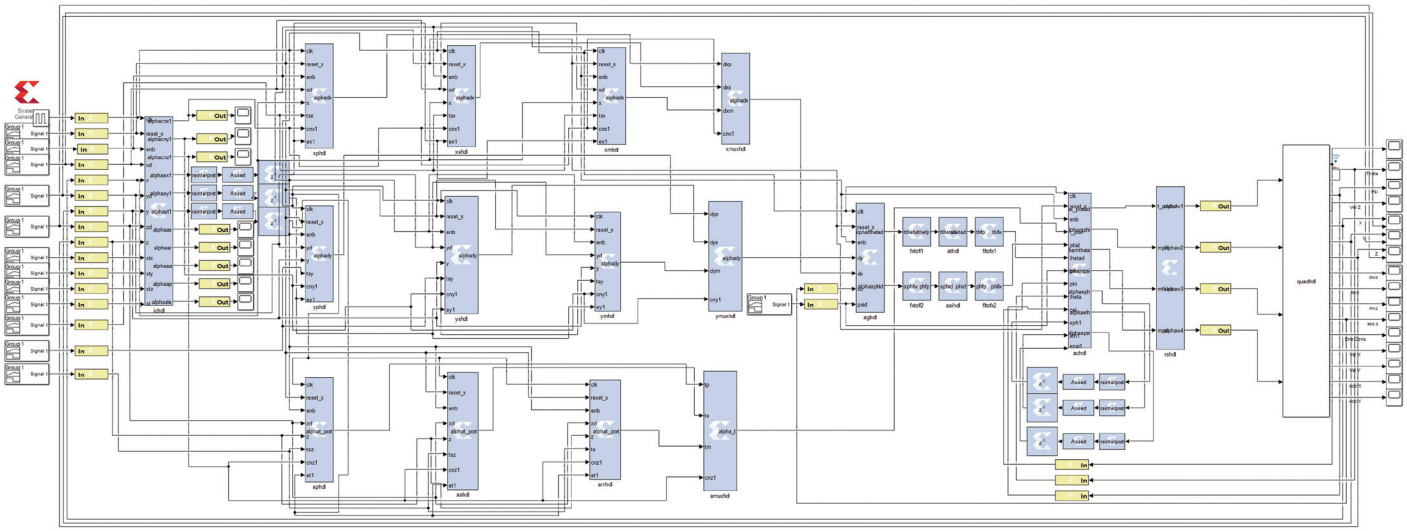

Figure 5. Complete switching based controller to be co-simulated with quadcopter model.

Table 3. Observations about speed, overshoot, and energy

Controller	Observed settling time (sec)	Per cent overshoot	Prop. energy (KJ)	Assigned ranks for speed and energy	Ranks complied
PD	5.795	5.5	15.5	3	Yes
SMC	4.377	0	15.29	2	Yes
MPC	4	3.6	15.21	1	Yes

Table 4. Implementation summary for switching/DPR-based controllers

Switching/ DPR based	Modules	Net power W	LUT 47232	FF 94464	DSP 240	IOB 252	Area and reliability ranks assigned	Ranks complied
Switching	Complete/Top	0.315	28987	366	212	200	--	--
DPR	SMC + Static	0.297	19341	240	172	200	2	Yes
	MPC + Static	0.295	19108	227	152	200	3	Yes
	PD + Static	0.275	9560	113	120	200	1	Yes
	Only static	--	9516	75	116	--	--	--

Table 5. Per cent savings by DPR-based approach over switching-based approach

RMs	LUTs saved (%)	FFs saved (%)	DSPs saved (%)	Computation power saved (%)
SMC	33	34	18	5.7
MPC	34	38	28	6.3
PD	67	69	43	12

Co-Simulation of FPGA-based position controller controlling Simulink Quadcopter model (based on System Generator) as shown in Fig. 5.

8. IMPLEMENTATION RESULTS AND ANALYSIS

In Table 4, as SMC uses resources only for a small portion of the tracking period, unlike MPC, as stated in Section III, its ranks for area and reliability assigned stand complied.

8.1 Summary of Results

- The switching-based controller can optimise the tracking speed, reliability, and propulsion energy metric, as validated by the co-simulation results of the previous section (Table 3).
- It occupies a large controller area as inactive controllers are also to be placed (Table 4).
- The computational power is moderate, as many controllers are off (Table 4).
- DPR is the best way to ensure complete optimization with improvements in controller area and computation power (Table 4).

Table 5 summarises the per cent improvements in each resource and computation power with SMC/MPC/PD as the Reconfigurable Modules (RMs) along with static modules compared to the switching-based controller.

8.2 Co-Simulation of the Position Controller for Optimised Tracking Mode

Here, the controller is the same as that shown in Fig. 5,

Table 6. Input specifications

Iteration	Direction	Controller activated	Input, rate	Start time of tracking (sec)	Desired settling time (sec)
1	X	PD	5, 1.2	1	4
2	Y	SMC	5, Step	1.1	4
3	Z	MPC	5, Step	1.2	4
4	X	PD	5, 1.2	8	4
5	Y	SMC	5, Step	8.1	4
6	Z	MPC	5, Step	8.2	4
7	X	PD	5, 1.2	15	4
8	Y	SMC	5, Step	15.1	4
9	Z	MPC	5, Step	15.2	4

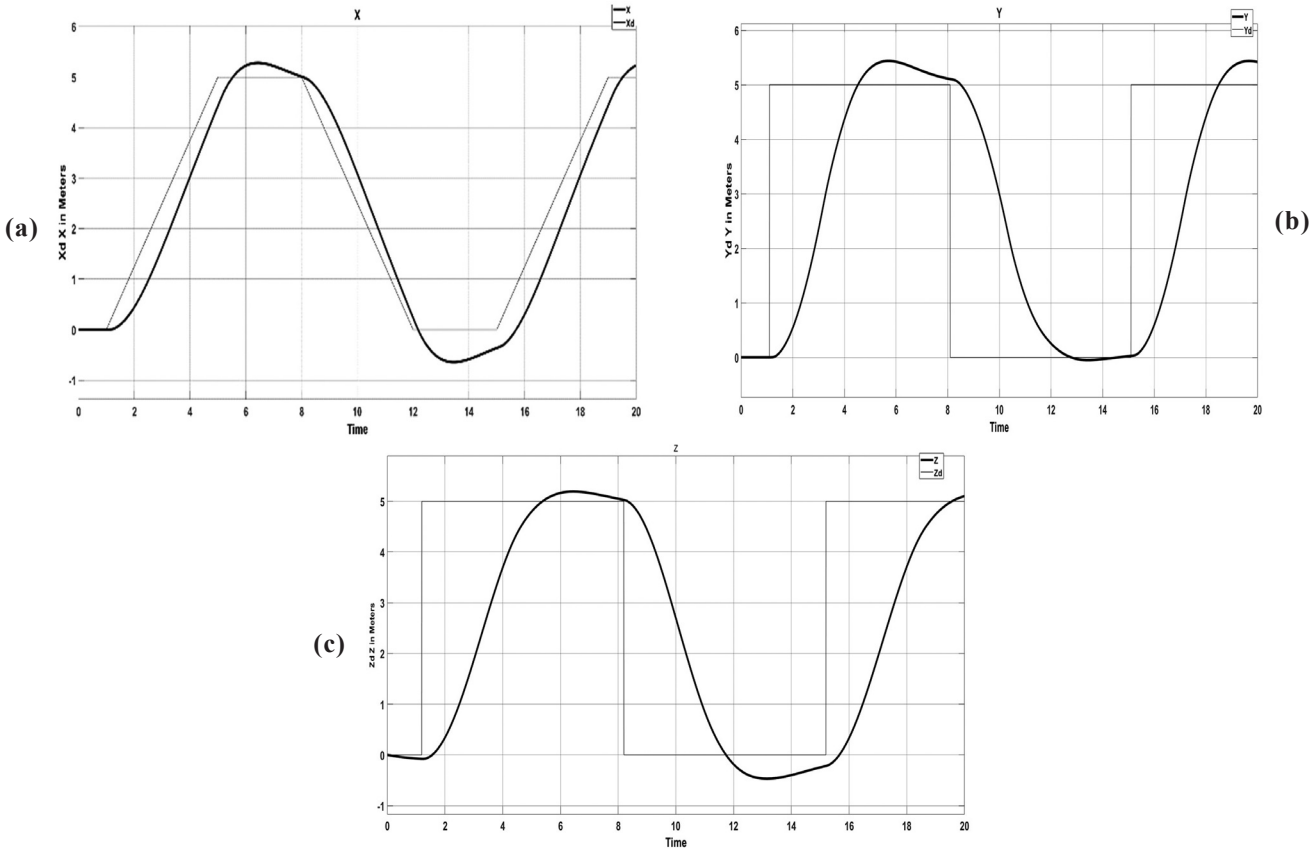

Figure 6. X tracking by PD; (b) Y tracking by SMC; and (c) Z tracking by MPC.

but with ptm=0, with the intelligent control activation block operating in the optimal mode for balanced tracking. The control task is to move the quadcopter from (0, 0, 0) to (5 m, 5 m, 5 m), (1 iteration each for X/Y/Z), back to (0, 0, 0), (next sets of iterations each for X/Y/Z) and again to the same (5 m, 5 m, 5 m), (last iterations each for X/Y/Z) as in Table 6. It is ensured that Y is applied after X and Z. Figure 6(a), Fig. 6(b), and Fig. 6(c) show corresponding trackings in the X/Y/Z directions. In total, there are 9 iterations for X/Y/Z combined. Hence, in the end, the best ranks are 9, and the worst ranks are 27, and the intelligent control activation block ensures average cumulative equal ranks of 18 for each of the speed, reliability, area, and power metrics, giving importance to the optimality of each of these metrics, as observed in Fig. 7.

8.3 Results Analysis

The overshoot and actual settling time columns in Table 7 indicate that the controller reasonably meets the desired control objectives. Equal cumulative ranks observed here for the metrics at the end of each of the 3 iterations indicate that the controller within 3 iterations also ensures balanced tracking with reference to speed, reliability, area, and power.

9. CONCLUSIONS

The evolvable hardware-based position controller validated for performance has optimised speed and energy metrics, as verified by co-simulation. The Vivado-based implementation results also indicate the optimisation of resources/reliability and computational power with the DPR-based approach compared to the switching-based controller.

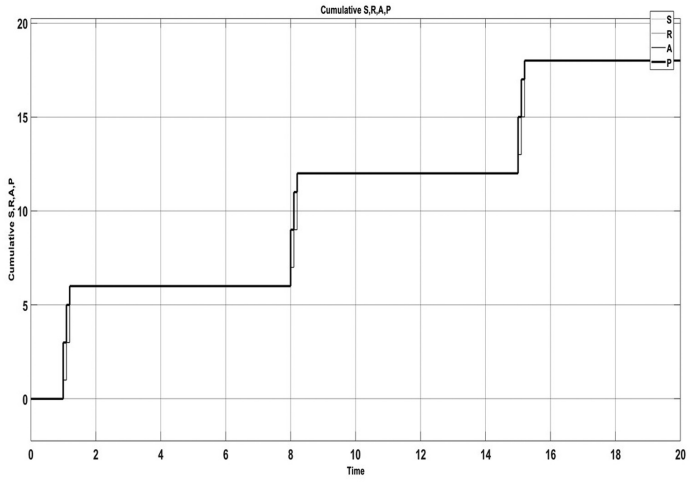

Figure 7. Cumulative S,R,A Pranks.

Table 7. Results of co-simulation in optimal tracking mode

Iteration	Direction	Controller activated	Per cent overshoot	Actual settling time sec	Cumulative ranks			
					Speed	Reliability	Area	Power
1	X	PD	5.6	5.82	3	1	1	3
2	Y	SMC	8.7	6.04	5	3	3	5
3	Z	MPC	3.8	4.4	6	6	6	6
4	X	PD	12.8	7	9	7	7	9
5	Y	SMC	1.05	4.02	11	9	9	11
6	Z	MPC	9.3	6.76	12	12	12	12
7	X	PD	4.5	5	15	13	13	15
8	Y	SMC	8.6	5	17	15	15	17
9	Z	MPC	1.8	4.4	18	18	18	18

Larger overshoots observed for the controllers may be improved with better/advanced controllers. Testing and reconfiguration have the potential to ensure the enhanced reliability of such complex systems. The FPGA device considered (xczu2cg-sfvc-784-2e) has an area of less than 6.25 cm^2 with enough resources still available for a DPR-based controller, which has the potential to implement all these improvements, unlike a switching-based controller.

REFERENCES

1. Trimeche, A., Sakly, A., Mtibaa, A. & Benrejeb M. PID controller using FPGA technology. Advances in PID Control, edited by Dr. Valery D. Yurkevich. *InTech*, 2011.
   doi: 10.5772/18295.
2. Somani, A. & Kokate, R. Realisation of FPGA based PID controller for speed control of DC motor using xilinx SysGen. Information, and Communication Technology for Intelligent Systems (ICTIS 2017) 2, Smart Innovation, Systems and Technologies., 2018, **84**.
   doi: 10.1007/978-3-319-63645-0_9.
3. Ozev, M.; Ortatepe, Z.; & Karaarslan, A. An FPGA-based comparative analysis of control techniques for gimbals and fins of missiles, *Electrica,* 2022, **22**(2), 160-172.
4. Sukavanam, N. & Felix Orlando, M. Sliding mode controller, Lectures 33 and 34, Robotics and Control, Theory and Practice, *NPTEL-Swayam online Course*, 26th Oct, 2021, IIT Roorkee, India.
5. Sulaiman, N.B.; Marhaban, M.H.; Piltan, F.; Nowzary, A. & Tohidian M. Design of FPGA-based Sliding Mode Controller for robot manipulator. *Int. J. Robot. Autom.,* 2011, **2**(3), 183-204.
6. Rajagopal, K.; Laarem, G. & Karthikeyan, A. FPGA implementation of adaptive sliding mode control and genetically optimized PID control for fractional-order induction motor system with uncertain load. *Adv. Differ. Equ.,* 2017, **273**.
7. Patwardhan, S.C. Model predictive control, Lectures 25 and 26, Advanced Process Control, *NPTEL Video Lectures*, 17th Jan 2022, IIT Bombay, India.
8. Lee, J. & Chang, H.J. Analysis of explicit model predictive control for path-following control. *PLoS ONE.*, 2018 **13**(3), e0194110.
   doi: 10.1371/journal.pone.0194110.
9. Ozan, G. & Mustafa, G. Efficient FPGA-based real-time implementation of model predictive control for single-phase direct matrix converter. *Electr. Power Compon. Syst.*, 2020, **48**, 16-17, 1773-1785.
   doi: 10.1080/15325008.2021.1908448.
10. Szolc, H. & Tomasz, K. Hardware-in-the-loop simulation of a UAV autonomous landing algorithm implemented in SoC FPGA. *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*., 2022, 135-140.
11. Biswas, S. Weeks 1 to 4, Optimisation Techniques for Digital VLSI Design, *NPTEL Video Lectures*, 25th June 2022, IIT Guwahati, India.
12. Hadidi, R.; Asgari, B.; Jijina, S.; Amyette, A.; Shoghi, N. & Kim, H. Quantifying the design-space trade-offs in autonomous drones. Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21), 19-23 April 2021, Virtual, ACM, New York, NY, USA, 2021.
    doi: 10.1145/3445814.3446721.
13. Cao, Y.; ElMaraghy, H.A. & Azab, A. Reconfigurable control structure for robots in assembly. *J. Intell Robot.*

Syst., 2007, **50**, 419–439.
doi: 10.1007/s10846-007-9173-6.
14. Emanuele, S.; Tatiana, K. & Cyrille, L. A novel genetic algorithm for evolvable hardware. *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 1st Jan, 2006, 134-141.
doi: 10.1109CEC.2006.1688300.
15. Konomura, R. & Koichi, H., Phenox: Zynq 7000 based quadcopter robot, *In* International Conference on ReConFigurable Computing and FPGAs (ReConFig14), 2014, pp.1-6.
16. Noe, M.; Bosco, D.; Salvador, E.; Montoya, J.; Jarquin, F. & Carlos, T. B. Design, development and implementation of a UAV flight controller based on a state machine approach using an FPGA embedded system". *IEEE/AIAA Digital Avionics System Conference (DASC)*, Sacramento, USA, 2016, 1-8.
doi: 10.1109/DASC.2016. 7778069.
17. UG904, Vivado Design Suite User Guide for Implementation, v2018.3.
18. Kovari, B.B. & Ebeid, E. MPDrone: FPGA-based platform for intelligent real-time autonomous drone operations. *IEEE* International Symposium on Safety, Security, and Rescue Robotics, New York City, NY, USA, 2021, 71-76,
doi: 10.1109/SSRR53300.2021.9597857.
19. Solovyov, A.M.; Semenov, M.E.; Selvesyuk, N.I.; Kosyanchuk, V.V.; Zybin, E.Y.; Glasov, V.V. Dynamic reconfiguration of a distributed information-computer network of an aircraft. *In* Taratukhin, V., Matveev, M., Becker, J., Kupriyanov, Y. (eds) Information Systems and Design. ICID 2021. *Commun. Comput. Inf. Sci.,* **1539.**
doi: 10.1007/978-3-030-95494-9_10.
20. Wan, Z.; Bo, Y.; Thomas, Y.L.; Jie, T.; Yuhao Z.; Wang, Y.; Raychowdhury, A. & Shaoshan, L. A survey of FPGA-based robotic computing, *IEEE Circuits Syst. Mag.* 21., 2020, 48-74.
21. Babu, P. & Eswaran, P. Reconfigurable FPGA Architectures: A survey and applications. *J. Inst. Eng. (India) Series B.,* 102, 2022, 143–156.
doi: 10.1007/s40031-020-00508-y.
22. Biruk, S.; Marco, P.; Alessandro, B. & Giorgio, B. Automating the design flow under dynamic partial reconfiguration for hardware-software co-design in FPGA SoC, *SAC'21.* Proceedings of the 36th Annual ACM Symposium on Applied Computing, March 2021, 481-490. doi: 10.1145/ 3412841.3441928.
23. Russell, T.; Kenneth, L.P. & Andre D. Reconfigurable computing architectures. Proceedings of the *IEEE.,* 2015, **103**, 332-354.
24. Bobda, C. Partial reconfiguration design, introduction to reconfigurable computing, ed. Bobda, C., *Springer, Dordrecht.,* 2007.
doi: 10.1007/978-1-4020-6100-4_7.
25. Kechiche, L.; Touil, L.; Jemai, M. & Ouni, B. A Power-aware real-time system for multi-video treatment on FPGA with dynamic partial reconfiguration and voltage scaling. *Eng. Technol. Appl. Sci. Res.,* 2022, **12**(4), 8996–9004.
26. Kidav, J.U. & Mohan, V. (2023). Design and development of an FPGA-based real-time reconfigurable computing platform. *In* Singh, S.N.; Mahanta, S. & Singh, Y.J. (eds) Proceedings of the NIELIT's International Conference on Communication, Electronics and Digital Technology. NICE-DT 2023. Lecture Notes in Networks and Systems. Singapore, 2023, **676**.
doi: 10.1007/ 978-981-99-1699-3_24.
27. UG909, Vivado Design suite user guide, Partial Reconfiguration, v2018.3.
28. Sharma, D. Weeks 7 and 8, Evolutionary computation for single and multiobjective optimisation. *NPTEL-Swayam online course*, 20th Jan 2023, IIT Guwahati, India.

CONTRIBUTORS

Mr Harish Bhat N. obtained his MTech degree in Digital Electronics and Communication from Visvesvaraya Technological University, Belgaum, India. He is currently pursuing the PhD degree from the Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. His current research interests include: Digital control system design, autonomous control, evolvable hardware, artificial intelligence for evolvable hardware, computer vision for state estimation, fault tolerant operation and power efficient algorithms for quadcopters.
Contribution in the current study: He carried out literature review, co-simulation-based performance validation. Implementation of the switching-based/DPR-based controller on FPGA, performance analysis and written the final manuscript.

Dr Shreesha Chokkadi is currently working as a Professor and Head, Department of Instrumentation and Control Engineering, Manipal Academy of Higher Education, Manipal. He obtained his PhD from IIT, Bombay in Systems and Control IDP. His research interests are Robust control, optimal control, control relevant identification, electrical measurement systems, energy and image processing.
Contribution in the current study: He supervised methodology and implementation, carried out algorithms validation, results analysis leading to system improvements. He is also responsible for reviewing the original manuscript review and suggested improvements and also responsible for corresponding with journal editorial team.

Mr Satish Shenoy B. is currently working as a Professor with the Department of Aeronautical and Automobile Engineering, Manipal Academy of Higher Education, Manipal.
Contribution in the current study: He conceptualised and designed the study, carried out results analysis, reviewed and edited the overall the manuscript.