

Current controller design based on FPGA

Do Van Can*

Faculty of Engineering and Technology, Quy Nhon University, Vietnam

Received: 12/11/2019; Accepted: 15/12/2019

ABSTRACT

In closed-loop control systems for motors, a current controller is usually designed as the inner loop while the others are the speed and the position controllers. Therefore, the current controller usually does not need boundary conditions. Most of these controllers are currently operated on software, resulting in lots of computational volumes, complex algorithms and slow responses. To overcome these drawbacks, the current controller should be hardened to increase the responsiveness and reduce the load on the CPU. The goal of this paper is to present the allowance of FPGA technology for users to design hardening of the flow controller regardless of the CPU scan cycles. Experimental and simulation results show that the design of current controllers based on FPGA response meets the response requirements.

Keywords: *Current controller, FPGA, motor control, hardware, software.*

*Corresponding author.

Email: dovancan@qnu.edu.vn

Thiết kế bộ điều khiển dòng điện dựa trên nền tảng FPGA

Đỗ Văn Cần*

Khoa Kỹ thuật và Công nghệ, Trường Đại học Quy Nhơn, Việt Nam

Ngày nhận bài: 12/11/2019; Ngày nhận đăng: 15/12/2019

TÓM TẮT

Trong hệ thống điều khiển vòng kín cho động cơ, khâu điều khiển dòng điện thường được thiết kế là vòng trong cùng, bên ngoài là các vòng tốc độ và vị trí. Vì vậy, vòng điều khiển dòng điện đáp ứng đầu ra thường không cần các điều kiện biên. Hầu hết các bộ điều khiển hiện nay, ba vòng này được thực hiện trên phần mềm, dẫn đến khối lượng tính toán nhiều, thuật toán phức tạp và làm quá trình đáp ứng chậm. Do đó, bộ điều khiển dòng được thực hiện cứng hoá nhằm tăng khả năng đáp ứng và giảm tải cho CPU là mục tiêu bài báo này. Công nghệ FPGA cho phép người dùng có thể thiết kế cứng hóa bộ điều khiển dòng mà không phụ thuộc vào vòng quét CPU. Kết quả mô phỏng và thực nghiệm cho thấy bộ điều khiển dòng điện trên FPGA đáp ứng được yêu cầu đề ra.

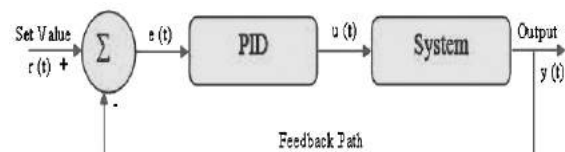
Từ khóa: Bộ điều khiển dòng, FPGA, điều khiển động cơ, phần cứng, phần mềm.

1. ĐẶT VẤN ĐỀ

Trong các hệ truyền động điện, bộ điều khiển dòng điện được thiết kế ở vòng trong cùng, bên ngoài là các mạch vòng tốc độ và vị trí.¹⁻³ Trước đây điều khiển dòng điện phần lớn được thực hiện bằng các thuật toán phần mềm dựa trên lỗi vi xử lý.^{1,2,4} Trong những năm gần đây, FPGA đã được ứng dụng vào lĩnh vực điều khiển. Nhiều nghiên cứu đã sử dụng FPGA nhằm giải quyết các yêu cầu tốc độ, tính đồng thời cho các cấu trúc điều khiển.⁵⁻⁷ Các công trình nghiên cứu trên cho thấy vẫn còn vài hạn chế trong điều khiển như thuật toán phức tạp, xử lý nhiều vòng lặp, chất lượng điều khiển phụ thuộc vào thuật toán lập trình và không cho phép thay đổi tái cấu trúc các thuật toán mới trên bộ điều khiển hiện có. Bên cạnh đó FPGA bước đầu sử dụng trong nghịch lưu đa bậc,⁸ điều khiển hiện đại ADRC⁹ và nhiều ứng dụng khác...^{6,10,11,12} Các kết quả mô phỏng của các bài báo trên cho thấy bước đầu đưa FPGA vào ứng dụng trong điều khiển. Kết

quả mô phỏng¹³ cho thấy hiệu quả tốt khi đề xuất FPGA thay cho DSP. Còn tài liệu¹² cũng hướng đến sử dụng FPGA cho điều khiển động cơ. Tuy nhiên, việc lựa chọn vòng điều khiển nào trên phần cứng, phần mềm để mang lại hiệu quả nhất; hay các kết quả thực nghiệm trên phần cứng chưa được đề cập cụ thể trong các tài liệu trên.

Để tiếp tục triển khai hướng đi của các công trình trên, tác giả đề xuất thiết kế bộ điều khiển dòng điện trên phần cứng dựa trên nền tảng FPGA nhằm tăng khả năng đáp ứng dòng điện và giảm tải cho CPU khi thực hiện điều chỉnh tự động.¹⁴ Dựa trên kiến trúc khâu điều khiển dòng điện là PID được sử dụng trong các bộ điều khiển hiện nay mô tả như **Hình 1**:¹⁵⁻¹⁸



Hình 1. Đề xuất cấu trúc khâu điều khiển dòng điện theo tài liệu¹⁷

*Tác giả liên hệ chính.

Email: dovancan@qnu.edu.vn

2. NỘI DUNG NGHIÊN CỨU

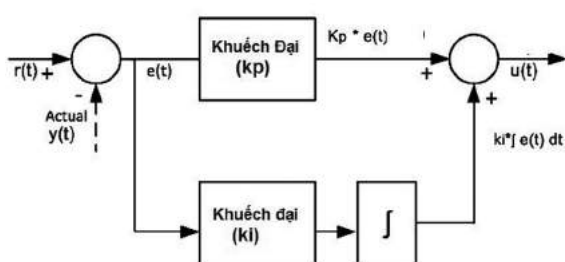
2.1. Đề xuất khâu điều chỉnh dòng điện

Cũng như những công trình trước đây khâu điều khiển dòng điện thường được lựa chọn PI (1) và chúng được thiết kế bằng thuật toán mềm.^{1,15} Trong bài viết này, tác giả sử dụng nguyên lý phương trình sai phân để xây dựng khâu điều khiển dòng điện PI (**Hình 2**) từ mô tả (1):

$$u(t) = k_p * e(t) + k_i * \int_0^t e(\tau) d\tau \quad (1)$$

Trong đó:

Sai lệch $e(t) = e(t) = r(t) - y(t)$



Hình 2. Sơ đồ khối khâu PI thực hiện theo phương trình sai phân (1)

Thực hiện số hóa trong miền z của phương trình (1) ta có:

$$U(z) = \left[k_p + \frac{k_i}{1 - z^{-1}} \right] E(z) \quad (2)$$

$$U(z) - z^{-1}U(z) = [k_p - z^{-1}k_p + k_i]E(z) \quad (3)$$

Phương trình sai phân của bộ điều khiển như sau:

$$u(k) = u(k-1) + k_p e(k) - k_p e(k-1) + k_i e(k) \quad (4)$$

Như vậy, khâu PI số được nhiều công trình nghiên cứu sử dụng làm bộ điều khiển dòng điện trong các ứng dụng phần mềm và mô tả theo phương trình sai phân (4). Vì vậy, cấu trúc này được lựa chọn thiết kế bộ điều khiển dòng điện trên phần cứng của FPGA cho động cơ.

2.2. Xây dựng khâu điều chỉnh dòng điện

Áp dụng cho hệ thống điều khiển dòng điện từ phương trình tổng quát PID số (5).^{4,5}

$$D(z) = \frac{U(z)}{E(z)} = K_p \left[1 + \frac{T}{\tau_i (1 - z^{-1})} + \tau_d \frac{(1 - z^{-1})}{T} \right] \quad (5)$$

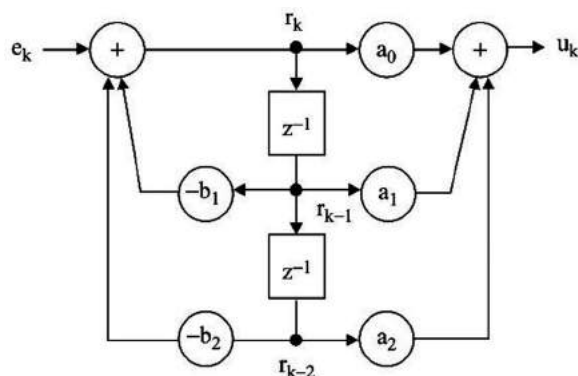
Viết lại dưới dạng (6) và mô tả **Hình 3**:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (6)$$

Trong đó:

$$a_0 = K_p \left(1 + \frac{T}{\tau_i} + \frac{\tau_d}{T} \right); \quad a_1 = K_p \left(1 + \frac{2\tau_d}{T} \right);$$

$$a_2 = K_p \frac{\tau_d}{T}; \quad b_1 = -1; \quad b_2 = 0$$



Hình 3. Sơ đồ khối khâu PID số tổng quát

Từ (4), ta có phương trình sai phân tổng quát sau:

$$u(k) = u(k-1) + a_0 e(k) + a_1 e(k-1) + a_2 e(k-2) \quad (7)$$

Thực hiện tương tự cho các cấu trúc điều khiển kinh điển.

$$D(z) = \frac{U(z)}{E(z)} = \frac{\sum_{j=0}^n a_j z^{-j}}{1 + \sum_{j=1}^n b_j z^{-j}} \quad (8)$$

$$u(k) = \sum_{j=0}^n a_j e(k-j) - \sum_{j=1}^n b_j u(k-j) \quad (9)$$

$$\text{Đặt: } M_1 = -b_1 r(k-1) - b_2 r(k-2);$$

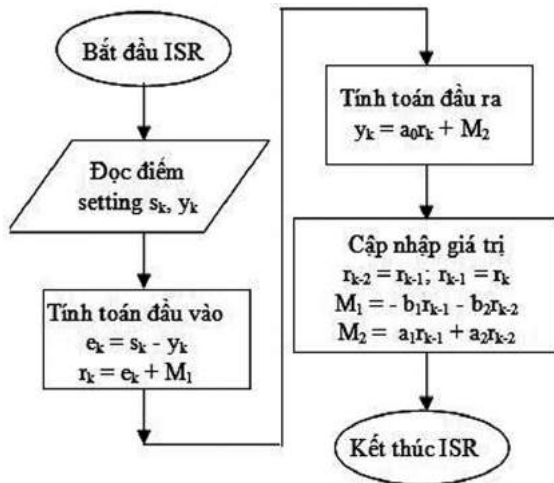
$$M_2 = a_1 r(k-1) + a_2 r(k-2).$$

Khi đó:

$$r(k) = e(k) + M_1 \quad (10)$$

$$u(k) = a_0 r(k) + M_2 \quad (11)$$

Và lưu đồ thuật toán tổng quát khi lập trình (chương trình ISR trên cả phần mềm và phần cứng) được mô tả như **Hình 4**. Như vậy, thuật toán này cho phép áp dụng cho mọi cấu trúc điều khiển số.



Hình 4. Sơ đồ thuật toán bộ điều khiển PID số

2.3. Biểu diễn hệ thống điều khiển số trên phần mềm

Để diễn tả một cách cụ thể, tác giả đề xuất bài toán tổng quát với hàm truyền số bậc hai của đối tượng, có các hệ số:

$$a_0 = 2.38 \cdot 10^{-5}, a_1 = 4.76 \cdot 10^{-5}, a_2 = 2.38 \cdot 10^{-5}; \\ b_1 = 1; b_2 = -1.903; b_3 = 0.9048.$$

Khi đó, hàm truyền số được mô tả như sau:

$$D(z) = 10^{-5} \cdot \frac{2.38z^2 + 4.76z + 2.38}{z^2 - 1.93z + 0.9048} \quad (12)$$

Ta có hàm truyền vòng hở với bộ điều khiển PID số như sau:

$$H(z) = \frac{U(z)}{E(z)} = G_P + G_I \frac{z+1}{z-1} + G_D \frac{z-1}{z-C} \quad (13)$$

Hay

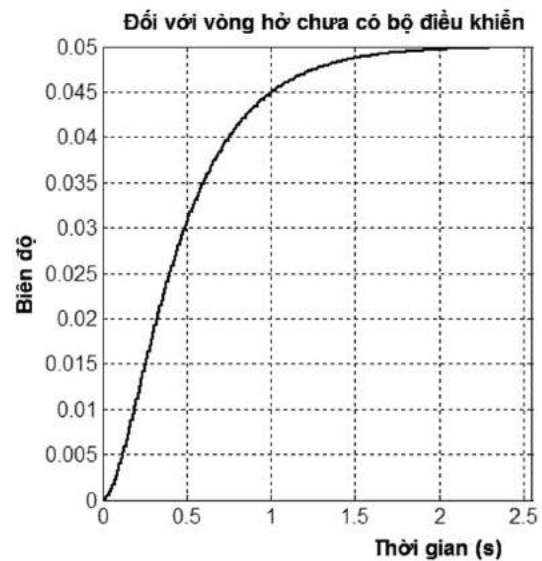
$$H(z) = \frac{U(z)}{E(z)} = G_P + G_I \frac{1+z^{-1}}{1-z^{-1}} + G_D \frac{1+z^{-1}}{1-Cz^{-1}} \quad (14)$$

Trong đó:

$$G_P = K_P; \quad G_I = K_I \frac{T_s}{2}; \\ G_D = \frac{2K_D}{T_s + 2T_F}; \quad C = \frac{T_s - 2T_F}{T_s + 2T_F}; \quad (15)$$

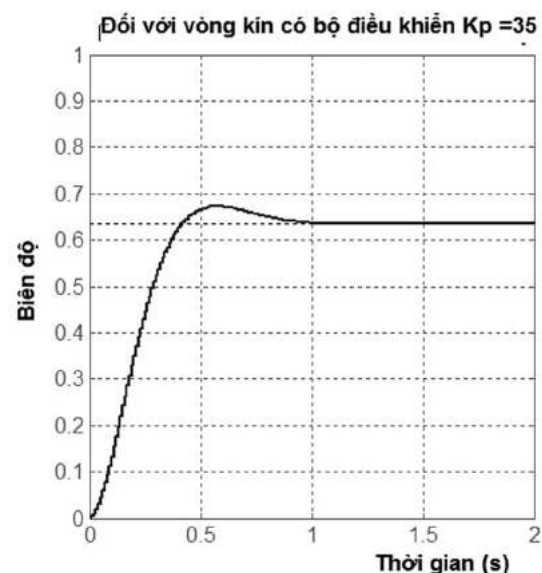
Trước tiên, chúng ta xây dựng bộ điều khiển dòng điện trên phần mềm Matlab cho đối tượng điều khiển trên. Bộ điều khiển số với chu kỳ trích mẫu $T_s = 0.1\text{ms}$, $T_F = 0.142$ kích thích đầu vào là hàm bước nhảy (Step), giá trị đặt là $r=1$.

Kết quả mô tả trên **Hình 5** là hệ chưa có bộ điều khiển số nên đáp ứng đầu ra 0.05/1 (đầu vào là 1) hệ chưa xác lập ổn định.



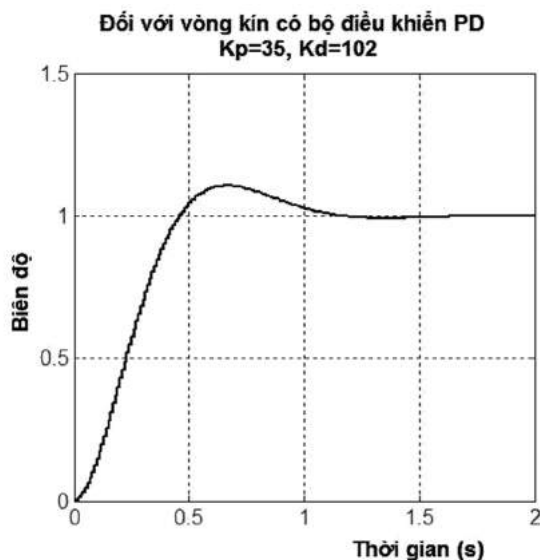
Hình 5. Kết quả mô phỏng khi chưa có bộ PID

Bộ điều khiển dòng điện cụ thể khâu P số được đưa thêm vào hệ thống, kết quả này được thể hiện trên **Hình 6**. Đáp ứng đầu ra 0.63 so với kích thích đầu vào 1, hệ chưa ổn định ở trạng thái xác lập.

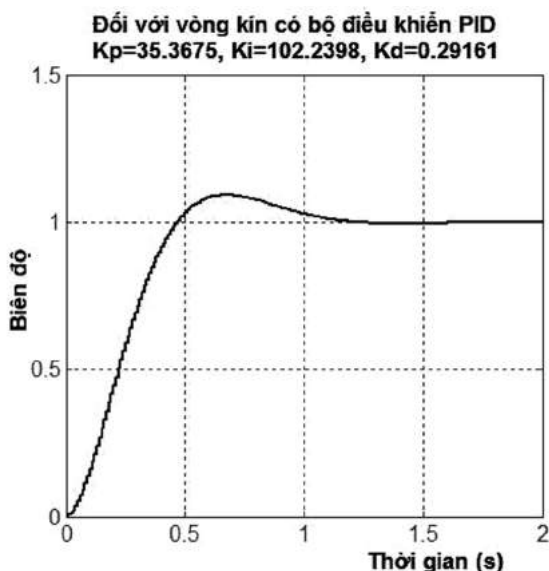


Hình 6. Kết quả mô phỏng khi sử dụng P

Tương tự, các bộ điều khiển PI, PID số cũng được thực hiện cho hệ thống này. Bằng kết quả tối ưu bộ điều khiển PID số trên Matlab, ta tìm được điểm ổn định cho hệ thống điều khiển **Hình 7** ứng với khâu PI số và **Hình 8** ứng với khâu PID số. Kết quả cả hai trường hợp đều ổn định xác lập tại thời điểm 1.6s và giá trị đạt được các hệ số $K_p = 35$; $K_i = 102$; $K_d = 0.29$; $T_s = 0.0001s$.



Hình 7. Kết quả mô phỏng khi sử dụng PI



Hình 8. Kết quả mô phỏng khi sử dụng PID

Nhận xét: Khi thực hiện thuật toán điều khiển (**Hình 4**) trên phần mềm, hệ thống điều khiển PID số với đối tượng tổng quát cho kết quả tốt, hệ thống đáp ứng ổn định khi tối ưu hóa các hệ số K_p , K_i , K_d .

Việc mô phỏng ở các chế độ của bộ điều khiển từ P, PI, PID để chỉ ra rằng bộ điều khiển dòng điện nằm trong cùng có thể thực hiện khâu PI vẫn có thể bảo đảm độ ổn định, rút ngắn thời gian thực hiện và giảm bộ nhớ so với bộ PID khi xây dựng trên phần cứng.

2.4. Xây dựng bộ điều khiển dòng điện trên phần cứng của FPGA

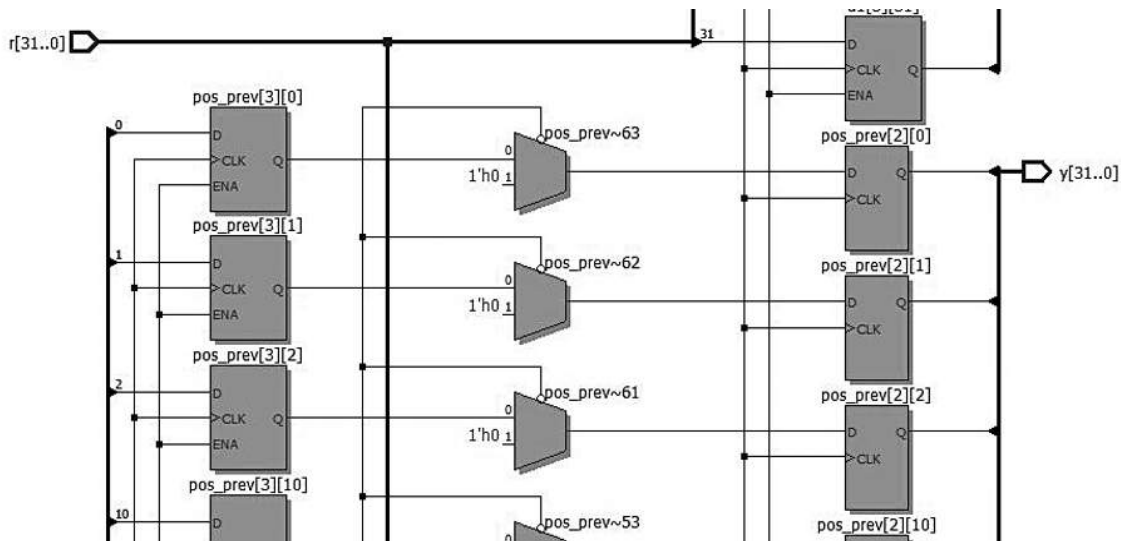
Như đã đề cập trong phần đầu, bộ điều khiển số được xây dựng trên nền tảng FPGA nhằm giảm tải vòng quét của CPU ở các thuật toán mềm, tăng khả năng đáp ứng cho các vòng điều khiển ngoài. Như đã diễn tả trên Matlab bộ điều khiển dòng điện chỉ cần khâu PI hoặc PID cũng mang lại hệ thống ổn định cho vòng trong. Vì vậy, khâu điều khiển dòng điện PI được thực hiện cứng hóa dựa trên nền tảng FPGA theo phương trình sai phân.

Từ (12), phương trình sai phân được mô tả trong trường số nguyên của FPGA:

$$10^7 y(k) = 1903.10^4 y(k-1) - 9048.10^3 y(k-2) + 238u(k) + 476u(k-1) + 238u(k-2) \quad (16)$$

$$\text{Trong đó: } e(k) = r(k) - y(k) \quad (17)$$

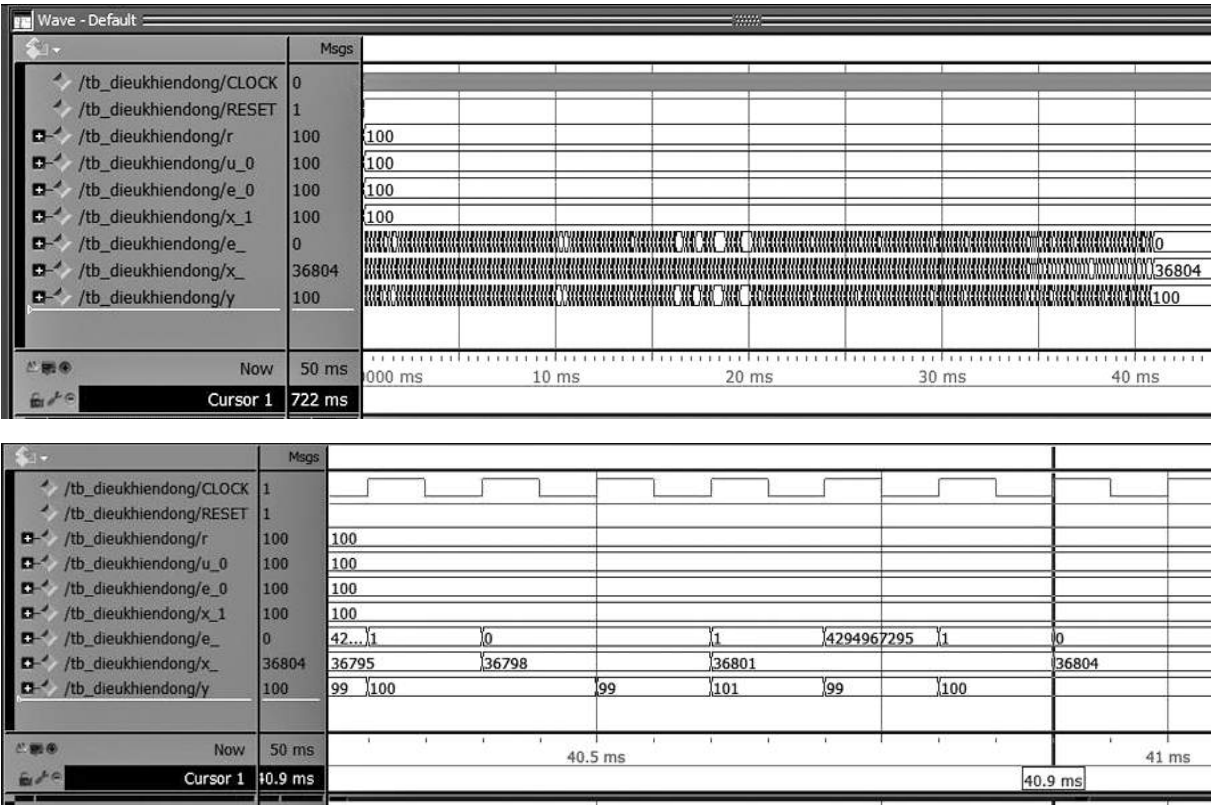
Trên **Hình 9** là một phần kết quả của hệ thống điều khiển xây dựng trên nền tảng FPGA dựa trên phương trình sai phân (16) gồm đối tượng điều khiển được cho trong (12) và bộ điều khiển số PI, ngôn ngữ lập trình phần cứng cho FPGA là Verilog.



Hình 9. Một phần cấu trúc hệ điều khiển sử dụng bộ PI số xây dựng trên nền tảng FPGA

Sau khi xây dựng hệ thống điều khiển với khâu điều khiển số PI cứng hóa trên FPGA, tác giả tiến hành mô phỏng kết quả trên phần mềm ModelSim của Altera với chu kỳ CLOCK là 0,1ms (trong thực tế hệ thống này cho phép mô phỏng lên đến 1ps), giá trị đặt $r = 100$ (số nguyên khi thực hiện trên chip). Kết quả được chỉ ra trên **Hình 10** (a-tổng thể, b là phóng lớn) cho thấy hệ

thống ổn định xác lập ở thời gian 41ms với giá trị đầu ra xác lập $y = 100$ tương ứng giá trị đầu vào đặt $r = 100$. Nếu so sánh với việc mô phỏng trên Matlab thì số chu kỳ để đạt được giá trị xác lập ổn định thì ở đây đáp ứng nhanh hơn nhiều. Điều đó nhờ vào hoạt động độc lập của bộ điều khiển được thiết kế cứng, chúng không phụ thuộc vào các vòng quét của CPU.

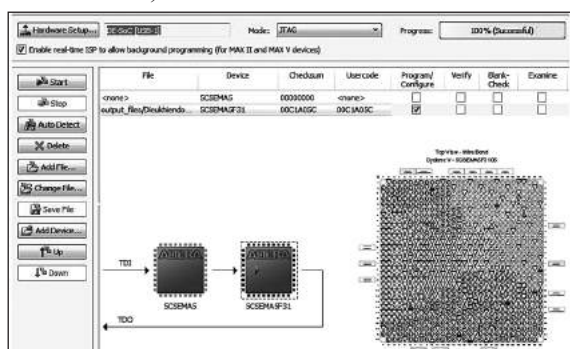


Hình 10. Kết quả mô phỏng hệ thống khi xây dựng bộ điều khiển số trên FPGA

Như vậy, ta thấy thực hiện bộ điều khiển trên phần cứng và phần mềm đều cho kết quả của hệ thống ổn định và xác lập trạng thái đáp ứng đầu ra. Đồng thời tốc độ làm việc của bộ điều khiển PI số được xây dựng trên phần cứng (Hardware) phụ thuộc vào tốc độ Clock của FPGA mà không phụ thuộc vào chu kỳ vòng quét của CPU như trước đây.

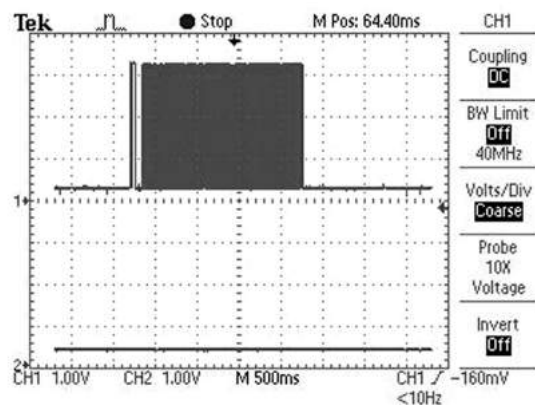
2.4. Xây dựng hệ thống thực nghiệm

Ưu điểm của FPGA trong việc thiết kế phần cứng một cách dễ dàng và khả năng tái cấu trúc phần cứng nhiều lần hay cho phép người dùng thay đổi cấu trúc bộ điều khiển ngay cả khi đang hoạt động. Trong nội dung mục này, tác giả thực hiện cấu hình thực nghiệm bộ điều khiển dòng điện (PI) trên nền tảng FPGA chip Cyclone V 5CSEMA5F31C6 của hãng Altera. Thực nghiệm này được thực hiện trên KIT De1SoC. Sau khi thiết kế bằng ngôn ngữ Verilog, ta tiến hành biên dịch để tạo file.sof chứa chuỗi Bitstream (nhị phân) cấu hình cho FPGA. Đối với KIT De1SoC cấu trúc 5CSEMA5F31C6 chia làm hai phần: một là lõi mềm ARM, hai là FPGA ở đây file.sof chỉ cấu hình cho phần FPGA (phần bên phải của Hình 11).

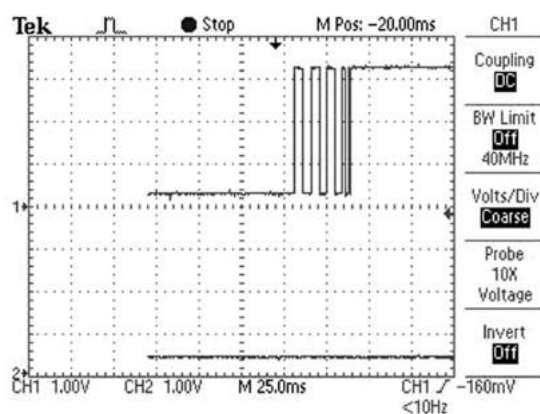


Hình 11. Cấu hình phần cứng trên chip Cyclone V 5CSEMA5F31C6 - KIT De1SoC Altera

Sau khi cấu hình thành công trên phần cứng FPGA, hệ thống được chạy thực trên KITDe1Soc và đo đạt đầu ra trên oscilloscope thông qua các chân đáp ứng đầu ra “y”. Kết quả Hình 12 là quá trình tổng thể và thời điểm quá trình quá độ Hình 13.

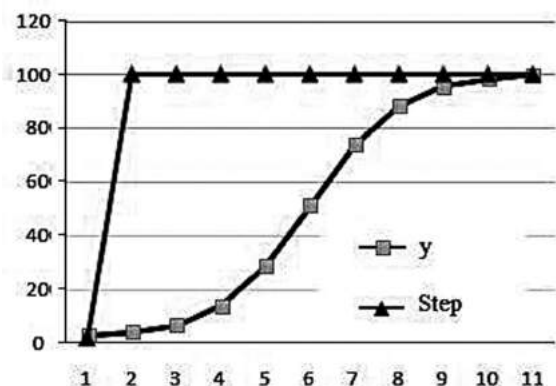


Hình 12. Kết quả thực nghiệm hệ thống điều khiển



Hình 13. Quá trình quá độ của thực nghiệm

Để dễ dàng biểu diễn trạng thái đáp ứng đầu ra, ta tiếp tục tiến hành thay đổi tần số hiển thị biểu diễn các kết quả thực nghiệm khi truy xuất các số liệu thực nghiệm từ file.xls trên oscilloscope điện tử và xây dựng biểu đồ như Hình 14.



Hình 14. Biểu diễn hệ thống đáp ứng đầu ra “y” và kích thích đầu vào từ các số liệu thực nghiệm

Một lần nữa, kết quả thực nghiệm cho thấy việc xây dựng bộ điều khiển dòng điện trên nền

tảng FPGA là hoàn toàn phù hợp hướng nghiên cứu trong tương lai và tiến đến ứng dụng trên các thiết bị điều khiển thương mại.

3. KẾT LUẬN

Với nội dung thực hiện ở trên, bài báo đã đưa ra được các vấn đề sau đây: Đề xuất xây dựng bộ điều khiển dòng điện (cụ thể PI, hoặc PID) trên nền FPGA là hoàn toàn hợp lý. Việc xây dựng bộ điều khiển dòng điện trên phần cứng giúp giảm tải cho CPU và tăng khả năng đáp ứng, rút ngắn thời gian thực hiện điều khiển.

Bằng kết quả mô phỏng trên phần mềm ModelSim và thực nghiệm trên KITDe1SoC cho kết quả đáp ứng tốt, thời gian xác lập nhanh, không phụ thuộc vào chu trình phần mềm.

TÀI LIỆU THAM KHẢO

1. W. Wang and P. Zhao. Application of kalman filter in the CNC servo control system, *Symp. Secur. Detect. Inf. Process.*, **2010**, 7, 442-446.
2. Y. C. Xiaohong Wang, Zhiffeng Pan, Thi Thu Giang Hoang, Liafang Tian. New Repetitive current controller for PWM Rectifier, *IFAC International Fed. Autom. Control.*, **2018**, 154-159.
3. E. R. Alphonsus and M. O. Abdullah. A review on the applications of programmable logic controllers (PLCs), *Renew. Sustain. Energy Rev.*, **2016**, 60, 1185-1205.
4. P. Ponce, A. Molina, G. Tello, L. Ibarra, B. MacCleery, and M. Ramirez. Experimental study for FPGA pid position controller in CNC Micro-machines, *IFAC Proc. Vol.*, **2015**, 48(3), 2203-2207.
5. M. Aboelaze and M. G. Shehata. Implementation of Multiple PID Controllers on, *978-1-5090-0246-7/15/2015 IEEE*, **2015**, 446-449.
6. L. Kong, L. Zhu, L. Zhang, H. Bao, and C. Rao. Real-Time Controller Based on FPGA and DSP for Solar Ground Layer Adaptive Optics Prototype System at 1-m NVST, *IEEE Photonics J.*, **2017**, 9(2), 1-11.
7. B. D. Reddy, N. K. Anish, M. P. Selvan, and S. Moorthi. Embedded control of n-level DC-DC-AC inverter, *IEEE Trans. Power Electron.*, **2015**, 30(7), 3703-3711.
8. J. Karttunen, S. Kallio, J. Honkanen, P. Peltoniemi, and P. Silventoinen. Stability and performance of current harmonic controllers for multiphase PMSMs, *Control Eng. Pract.*, **2017**, 65, 59-69.
9. M. R. Stanković, S. M. Manojlović, S. M. Simić, S. T. Mitrović, and M. B. Naumović. FPGA system-level based design of multi-axis ADRC controller, *Mechatronics*, **2016**, 40, 146-155.
10. Q. Tang, L. Jiang, X. X. Liu, and Q. Dai. A real-time updatable FPGA-based architecture for fast regular expression matching, *Procedia Comput. Sci.*, **2014**, 31(2014), 852-859.
11. J. V. Frances-Villora, A. Rosado-Muñoz, J. M. Martínez-Villena, M. Bataller-Mompean, J. F. Guerrero, and M. Wegrzyn. Hardware implementation of real-time Extreme Learning Machine in FPGA: Analysis of precision, resource occupation and performance, *Comput. Electr. Eng.*, **2016**, 51, 139-156.
12. Y. Chen, B. Xie, and E. Mao. Electric Tractor Motor Drive Control Based on FPGA, *IFAC-PapersOnLine*, **2016**, 49(16), 271-276.
13. M. Curkovic, K. Jezernik, and R. Horvat. FPGA-based predictive sliding mode controller of a three-phase inverter, *IEEE Trans. Ind. Electron.*, **2013**, 60(2), 637-644.
14. J. Cigánek, M. Kocúr, and Š. Kozák. Hardware Realization of Advanced Controller Design Methods using FPGA, *IFAC-PapersOnLine*, **2016**, 49(5), 163-168.
15. M. Hanifzadegan and R. Nagamune. Contouring Control of CNC Machine Tools Based on Linear Parameter-Varying Controllers, *IEEE/ASME Trans. Mechatronics*, **2016**, 21(5), 2522-2530.
16. Nguyen Phung Quang. *Điều khiển tự động truyền động điện xoay chiều 3 pha*, Nxb Giáo Dục, 1998.
17. V. B. Patel and R. H. Acharya. Design of FPGA-based All Digital PID Controller for Dynamic Systems, *Int. J. Adv. Res. Electr. Electron. Instrum. Eng. Vol. 1, Issue 2*, **2012**, 1(2), 64-72.
18. G. Zhao, Y. Zhao, A. Dong, and L. Zhang. Research on Nonlinear PID Position Controller of CNC System, *2007 IEEE Int. Conf. Autom. Logist.*, **2007**, 2446-2450.