Microprocessor-based performance indices analysis of individual systems distributed motion control strategies

Wasim Ghder Soliman^{1,2} · D. V. Rama Koti Reddy² · D. Akhila Reddy³

Received: 21 August 2019 / Accepted: 30 November 2019 / Published online: 5 December 2019
© Springer Nature Switzerland AG 2019

Abstract

This paper objective is to analyze individual systems performance indices of a developed synchronization control for bi-axial system speed control/speed synchronization by using two strategies (cross-coupling technique and master–slave technique) with two permanent magnet direct current (PMDC) motors. Two prototypes of the previously developed microprocessor-based PMDC motor are used. The control algorithms are implemented using ARDUINO Mega board after installing MATLAB supporting package for it. First, the system identification procedure is carried out using a pseudo-random binary sequence test and the transfer function is derived. Then, optimal tuning of the Proportional-Integral controller is performed based on the multiple-application Simpson's 1/3 rule with the performance indices as objective functions of the tuning algorithm. After that, both cross-coupling and master–slave speed synchronization algorithms are implemented. Finally, a comparison is made for their performance indices. The integral absolute error, integral square error, integral time absolute error and integral time square error performance indices are used for the comparison purpose. Introducing the analysis of the performance indices values for individual systems is the main contribution of this paper. Larger values of the performance indices resulted in master–slave strategy on slave PMDC motor due to the time delay needed for this motor to follow the master speed.

Keywords System identification · Controller design · Cross-coupling strategy · Master–slave strategy · Integral of absolute error (IAE) · Integral square error (ISE) · Integral time absolute error (ITAE) · Integral time square error (ITSE) · Real-time

1 Introduction

Synchronization Control of multiple numbers of motion axes or motors is one of the basic problems in motion control. Significant changes have been brought to motion control with the development of microprocessors, micro-computers and microcontrollers as they enable implementation of complex and sophisticated control techniques [1]. Due to their high performance and easy controller design, Direct Current (DC) motors are being used in automotive, vehicles and many other industrial applications [1].

In addition to that, DC motors have important applications in the academic world. With the help of a developed virtual laboratory, both the permanent magnet DC (PMDC) motor and the controller parameters are made changeable and this enables obtaining effective learning and comprehension of both Artificial Neural Networks (ANN) controllers [2] and Fuzzy Logic Controllers (FLC) [3]. PMDC motors are highly efficient, compact in size and robust [4], because they do not require any extra dc supply to generate the magnetic field. However, the PMDC motors big thrust force, high speed, and high precision [5–7] ensure

✉ Wasim Ghder Soliman, wasimsoliman1987@gmail.com; D. V. Rama Koti Reddy, rkreddy_67@yahoo.co.in; D. Akhila Reddy, akhilar3333@gmail.com | ¹Industrial Automation Engineering Department, Technical Engineering College, University of Tartous, Tartous, Syria. ²Department of Instrument Technology, College of Engineering, Andhra University, Vishakhapatnam, AP 530003, India. ³Department of Electronic and Communication Engineering, College of Engineering, Andhra University, Vishakhapatnam, AP 530003, India.

the expectation of their usage in many industrial applications. For good quality of the product, communication between the subsystems in the product production line is desired [8]. Speed synchronization has many applications in process control systems. In the textile industry, as an example of the automation industry, both let-off and take-up systems should be synchronized electronically for the aim of avoiding leaving starting marks in the fabric through ensuring a dynamic cloth correction after each machine stop [9]. Multi-axes motion control is very necessary. In order to reach consensus in multi-robot system, the synchronization is the key requirement [10]. Automation merges the different fields of technology and leads to the very advanced industry. In reality, we may need to synchronize n motor in order to obtain the final product in the industry. As the development is done with the help of computer and embedded systems, working with the digital domain is the basic role for designing any controller.

1.1 Related work

Adaptive feedforward controllers are designed and implemented for speed synchronization for two motion axes [11]. Fuzzy logic coupling control is described with the addition of linguistic inference outputs to each individual control loop [12]. A neuro-controller synchronization is introduced which results in better synchronization performance by using a new cost function and adjusting the neural network weights with the help of a learning law [13]. A new adaptive coupling-control algorithm is proposed for positions synchronize of manipulators along with its implementation on multirobot assembly systems [14]. Both coupling and synchronization factors are introduced into the definition of the synchronization error to result in an optimal synchronization control for motion systems [15]. A brief overview and explanation of several synchronization techniques are existed [16]; in addition to that, comparisons between those techniques are performed in both simulation and hardware implementation. A cross-coupling motion controller is described with introducing the most significant error as control design objective [17]. A simulation demonstration of an adaptive synchronization controller is presented [18], with a coupling controller for better synchronization between two motion axes. A robust cross-coupling controller is introduced which ensures contouring performance [19]. A synchronization algorithm is explained in a convenient way [20]; it is originally based on the cross-coupling methodology with a modification of introducing a synchronization factor. However, introducing this synchronization factor results in a more accurate cross-coupling especially when the working conditions require different trajectory tracking. The master–slave technique is implemented based on

a DSP board for the aim of speed synchronization of two DC motors [21]; similarly both simulation and hardware implementation are performed. Recently, the networked structures replace the classical automation pyramid [22] with the ability of data collection and analysis within cloud level. For the aim of flexible assembly manipulation in a dexterous way, a reconfigurable gripper with high-speed and multi-fingers is designed [23] with high force of movement and larger number of degrees of freedom (DoFs).

The common drawback of the related work is that; none of the previous work introduces the concept of performance indices of the used controllers for individual systems and this motivated us to use the individual controller's performance indices as points of comparison between cross-coupling technique and master–slave technique. A microprocessor-based embedded system (i.e. ARDUINO MEGA Board) is used for the hardware validation of the simulation work. Analyzing the controller's performance indices is aimed to come out with the speed synchronization scheme with optimal behavior for each individual system. This ensures the good working condition of the individual systems under the different working conditions. In addition to that, the recent direction in the industry is to use the state of the art technologies for remote control and observations, for example, the internet of things technology (IoT) integrates with the controllers to result in industrial IoT (IIoT), so current work is useful for IoT integration process [24] and can be extended for the same with optimal computations at edge level.

2 Work procedure

This work is consisted of two main sections; the first section is the software which is performed with the help of MATLAB SIMULINK for the simulation test and after that the installation of the ARDUINO support package that used for physical test. The second section is the hardware; it consists of two developed PMDC motors prototypes as actuators, including inbuilt driving circuits, and ARDUINO MEGA Board for algorithms implementation.

2.1 The utilized actuators and PRBS test for their system identification

The utilized actuators are two similar PMDC motors of the previously developed prototype [25]. However, the prototype contains five main parts, the first part is the PMDC motor kit with speed range from 0 to 1500 RPM, input Voltage range from 0 to 12 V_{dc} , and input current range from 0 to 1.5 A, the second part is the IR sensor unit with a slotted disk mounted on the front of the motor axis and Infrared light source and detector, the third part is the Frequency

Fig. 1 A microprocessor-based PMDC motor prototype

to voltage unit which provides, with the help of a microprocessor based inner circuit, a variable output direct voltage as the speed of the motor varies, the fourth part is the pulse width modulation (PWM) unit for interfacing the prototype with other microcontrollers for the aim of speed control and the fifth part is the Magnetic load mechanism for applying a magnetic linear load on the motor shaft. The above mentioned prototype along with its parts is illustrated in the Fig. 1.

The pseudo-random binary sequence (PRBS) test [26] is used for system identification procedure of the utilized actuators. The idea behind using the PRBS test is based on the fact of considering the physical working conditions while controlling the motor speed. The controller task is to maintain the output speed similar to the reference speed within different working conditions, i.e. load implementation or setpoint tracking, and this is performed through changing the provided PWM signals to motor armature

randomly according to the physical requirements. To this direction, a random set of input PWM signals are provided to the actuator and the output signals are observed. Both input/output signals are used for deriving the actuator model with the help of MATLAB system identification toolbox. However, the input/output signals are given/observed through developing a Simulink model with the help of MATLAB support package for ARDUINO hardware. As a result, the following transfer function is obtained by using sampling time equal to 0.001 s:

$$G_s = \frac{92.46s + 4.893}{s^2 + 124.2s + 4.173} \quad (1)$$

2.2 Multiple-application Simpson's 1/3 rule for the actuators controller tuning

In this section, the Proportional Integral (PI) controller is tuned for the above obtained model [given in Eq. (1)] based on the multiple-application Simpson's 1/3 rule [27, 28]. Simpson's 1/3 rule is one of the Newton–Cotes numerical integration schemes. It is based on the strategy of replacing tabulated data with an approximating function that is easy to integrate. Consider the following integration:

$$I = \int_a^b f(x)dx \cong \int_a^b f_n(x)dx \quad (2)$$

Where:

$$f_n(x) = aa_0 + a_1x + \dots + a_{n-1}x^{n-1} + a_nx^n \quad (3)$$

Where n is the order of the polynomial $f_n(x)$. When $n = 2$ i.e. $f_n(x)$ is a second-order polynomial, we get:

$$I = \int_a^b f(x)dx \cong \int_a^b f_2(x)dx \quad (4)$$

Let $a = x_0$, $b = x_2$ and

$$f_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_0) + \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_1) + \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_2) \quad (5)$$

(Lagrange Representation). Then we get:

$$I = \int_{x_0}^{x_2} \left[\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_0) + \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_1) + \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_2) \right] dx \quad (6)$$

The result after integration and manipulation will be:

$$I \cong \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] \quad (7)$$

(hence the name 1/3). Where

$$h = \frac{b-a}{2} \quad (8)$$

In order to improve Simpson's rule, the integration interval is divided into a number of equal width segments:

$$h = \frac{b-a}{n} \quad (9)$$

Then the total integration is given by the following formula:

$$I = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \dots + \int_{x_{n-2}}^{x_n} f(x)dx \quad (10)$$

Substituting Eq. (7) for the individual integral yields:

$$I \cong 2h \frac{f(x_0) + 4f(x_1) + f(x_2)}{6} + 2h \frac{f(x_2) + 4f(x_3) + f(x_4)}{6} + \dots + 2h \frac{f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)}{6} \quad (11)$$

Combining terms and using Eq. (8),

$$I \cong \frac{h}{3} \frac{f(x_0) + 4 \sum_{i=1,3,5}^{n-1} f(x_i) + 2 \sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)}{n} \quad (12)$$

(Multiple-Application Simpson's 1/3 rule).

The Multiple-Application Simpson's 1/3 rule is used to calculate the performance indices of the designed controller during the tuning procedure. The used performance indices are the integral absolute error (IAE), integral square error (ISE), integral time absolute error (ITAE) and integral

time square error (ITSE) and they are mathematically given by the following equations:

$$IAE = \int_0^{\infty} |e(t)|dt \quad (13)$$

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (14)$$

$$ITAE = \int_0^{\infty} t|e(t)|dt \quad (15)$$

$$ITSE = \int_0^{\infty} te(t)^2 dt \quad (16)$$

where $e(t)$ is the error signal and it is the difference between the desired speed and the output speed. How-

ever, the tuning procedure includes three main stages. In the first stage, a MATLAB Simulink model is developed in such a way that it is contained both PMDC motor derived transfer function and the PI controller blocks in terms of its proportional gain (k_C) and integral time constant (t_i) as it is illustrated in Fig. 2.

In the second stage, a MATLAB script is written in such a way that it contains the total simulation time, the initial estimated PI parameters values and the sampling time. In addition to that, the written script invokes the function "fminsearch" from MATLAB optimization tools. In the third stage, the objective function is written and it is contained a call to execute the developed Simulink model in the first stage and the code for Multiple-Application Simpson's 1/3 rule. The function "fminsearch" provides the initial k_C and t_i values to the objective function then the respective performance index is calculated on each evaluation of the objective function starting from the initial PI controller parameters values. The resulted k_C and t_i with the minimum performance indices values are listed in Table 1.

There are small differences between the values k_C and t_i with different performance indices and therefore an average value for both proportional gain and integral time constant are calculated and considered. In order to justify the designed PI controller, two physical tests are carried out. The first physical test is the disturbance rejection test in which a reference speed equal to 1000 RPM is provided

Fig. 2 The developed simulation diagram for the tuning procedure

Table 1 Tuned PI controller parameters using the performance indices

IAE tuning		ISE tuning		IATE tuning		ITSE tuning	
k_C	t_i	k_C	t_i	k_C	t_i	k_C	t_i
0.158	0.0037	0.150	0.005	0.16	0.004	0.155	0.003

Fig. 3 Disturbance rejection physical test **Fig. 4** Setpoint tracking physical test

to the controller which, in its term, maintains the output speed on this particular speed despite of the disturbances on the motor shaft as it is depicted on Fig. 3.

The second physical test is the setpoint tracking test and it is shown in Fig. 4, the reference speed is firstly set to be 900 RPM and the output speed follows this trajectory, then at physical time equal to 5 s. a new trajectory of 1100 RPM is applied and the controller makes the motor speed

Fig. 5 Cross-coupling strategy

to follow this trajectory, after that at test time equal to 10 s a new trajectory of 1300 RPM is applied and the controller makes the motor speed to follow this trajectory, finally and at test time equal to 15 s a new trajectory of 1000 RPM is applied and the controller makes the motor speed to follow this trajectory.

3 Speed synchronization of the two PMDC motors

Two methods are introduced to obtain the speed synchronization between the two PMDC motors. Those strategies are cross-coupling strategy and master–slave strategy. The aim is to compare the individual systems performance indices with both above-mentioned strategies in both simulation and real-time implementation.

3.1 Cross-coupling strategy

The coupling factor value should be equal to one when the speed synchronization is considered [20]. As the aim is to synchronize the speed, the utilized synchronization algorithm is modified and the coupling factor is eliminated. To this direction, the cross-coupling strategy is implemented

in such a way that the referenced trajectories i.e. r_1 and r_2 are provided directly to the cross-coupling block as it is shown in Fig. 5. For each individual system PI controller, a new signal is added; the new controller equation for first PMDC motor becomes:

$$u_1(t) = k_c e_1(t) + \frac{1}{t_i} \int e_1(t) + k_1 \quad (17)$$

And the new controller equation for second PMDC motor becomes:

$$u_2(t) = k_c e_2(t) + \frac{1}{t_i} \int e_2(t) + k_2 \quad (18)$$

where: k_1 and k_2 are the cross-coupling gains.

The synchronization factor i.e. $\frac{r_1}{r_2}$ is presented inside the cross-coupling block and this ensures the speed synchronization especially when working with different trajectories. In the practical situation, the proportional gain (k_c) and the integral time constant (t_i) are optimally obtained from the above mentioned tuning procedure, after that the value of k_1 is increased gradually from zero until it results in a minimum synchronization error, finally the value of k_2 is obtained through division of k_1 by the synchronization factor.

3.2 Master–slave strategy

In this method, one of the PMDC motors is selected to be the master motor and the reference speed is provided as input signal to its controller. However, the other motor is the slave motor as it is shown in the following control diagram (Fig. 6). The master motor output speed is the input signal of the slave motor controller. It is clear that there is no coupling between the two systems. With this strategy,

Fig. 6 Master–slave strategy

Fig. 7 Cross-coupling for speed synchronization

any change in the master motor speed is followed by the slave controller, whereas the changes in the slave motor speed cannot be seen by the master motor controller.

4 Simulation validation

After obtaining both PMDC motors models and PI controller parameters as mentioned above, the simulation procedure is developed using MATLAB SIMULINK environment. Then for the above-obtained results, the simulation test is carried on using two similar PMDC motors.

Fig. 8 Near view of applied load with cross-coupling method

Fig. 9 Master-slave for speed synchronization

Fig. 10 Near view of applied load with the master-slave method

4.1 Cross-coupling simulation test

Table 2 Performance indices simulation results

Performance index Simulation values	Cross-coupling		Master-slave	
	PMDC Motor 1	PMDC Motor 2	PMDC Motor 1	PMDC Motor 2
IAE	0.3703	0.3802	0.3703	0.3953
ISE	0.3210	0.3220	0.3085	0.3873
ITAE	0.3956	0.3857	0.3971	0.4152
ITSE	0.0800	0.0796	0.0756	0.0794

Figure 7 illustrates the cross-coupling strategy simulation response. The simulation test is performed for the cross-coupling of the PMDC motors and the above-mentioned synchronization algorithm is implemented between the two systems. Both systems were given a reference speed equal to 1000 RPM and the load was applied to first PMDC motor at simulation time 3.5 s and at simulation time 7 s on second PMDC motor. As it is observed the speed synchronization is obtained and in synchronization mode, the disturbance on any system will affect all the systems. If synchronization mode is not required, individual systems can be operated on individual speeds. Figure 8 shows a near view of the same response.

4.2 Master-slave simulation test

The master-slave simulation response is illustrated in Fig. 9. The simulation test is carried on for the work of the PMDC motors with a master-slave strategy for speed synchronization. The master system is given a reference speed equal to 1000 RPM and the load was applied on PMDC master motor at simulation time 3.5 s and at simulation time 7 s on PMDC slave motor. As it is clear from Fig. 10, the speed synchronization is obtained and the disturbance on the slave system will not affect the master system.

With this strategy, individual systems cannot be operated on individual speeds. In most industrial applications, the master motor is operated in open-loop mode and the controller is designed for the slave systems in order to follow its speed perfectly.

4.3 Summary of simulation results

The values of the performance indices are the points of comparison. The simulation results are summarized in Table 2. By analyzing the obtained results, it is clear that the performance indices values are larger with master-slave strategy slave motor. In fact, the slave motor takes a time delay to follow the master motor speed, and this time delay does not exist with cross-coupling strategy. This time delay increases the error area and causes larger values of the performance indices. Figure 11 shows the performance indices simulation values in a graph form.

Fig. 11 Performance indices simulation values in graph form

Fig. 12 Practical response of the coupled systems with load disturbance

5 Physical validation

The hardware implementation is required to justify the previously obtained simulation results. With the help of MATLAB SIMULINK and after installing ARDUINO support package for MATLAB SIMULINK, the hardware validation takes place.

5.1 Cross-coupling physical test

The hardware test is carried on for the work of the PMDC motors with a cross-coupling strategy for speed synchronization and the above-mentioned synchronization algorithm is implemented between the two motors. Both motors are given a reference speed equal to 1000 RPM and a load is firstly applied to first PMDC motor at the time 3.2 s

Fig. 13 A near view of load implementation with Cross_Coupling method

Fig. 14 Practical response of Master_Slave with the load

which affects the speed of the second motor as shown in Fig. 12. Similarly, a load is applied to the second PMDC motor at the time 6.2 s which affects the speed of the first motor. Figure 13 shows a near view of the same response.

It is observed that the speed synchronization is obtained and in synchronization mode, the disturbance on any system will affect other systems.

5.2 Master-slave physical test

The master-slave strategy hardware implementation is carried out for the work of the two PMDC motors for speed synchronization. The master motor is given a reference speed equal to 1000 RPM and the load is firstly applied

Fig. 15 Near view of load implementation with Master_Slave method

to master PMDC motor at the time 3.2 s which affects the speed of the slave motor as shown in Fig. 14. Similarly, a load is applied to the slave motor at the time 6.2 s which does not affect the speed of the master motor. The next figure (Fig. 15) illustrates how practically the PMDC slave motor follows the PMDC master motor trajectory in the near view version. The distance between the two speeds is an indication of the time delay required from the slave system to follow the master system.

5.3 Summary of physical results

The performance indices values are captured also for the aim of comparison between the two strategies and summarized in Table 3. The physical results match the previous obtained simulation results, it is clear that the performance indices values of the slave (second) PMDC motor of the master–slave method are larger and this is a reasonable result due to the existence of the delay time that the slave motor needs to follow the master motor speed, and this time delay does not exist with cross-coupling method. This time delay increases the error area and causes larger

values of the performance indices. Figure 16 shows the performance indices real-time values in a graph form.

It is clear that the values of the performance indices are increased by comparing both simulation and practical (real-time) results. This is due to the difference between simulation and hardware cases. The simulation case represents an ideal situation wherein real-time hardware implementation, and as the physical world is faced, it is reasonable to get such difference.

Actually, the simulation tests are based on the estimated models of the utilized actuators. In fact, the estimated transfer functions of the PMDC motors do not represent the exact model whereas the physical test represents the real behavior of the actuators and hence such different results between simulation and physical tests are obtained. For example, if another method of system identification is used or even if another set of the given random signals through the PRBS test are applied on the same actuator, then this will result in a different estimated mathematical model. So, the exact actuator behavior is observed with the physical test only and the obtained results are larger than the ideal (simulation) results. In addition to that, the applied load in the simulation tests is an ideal speed reduction of around 200 RPM, as it was illustrated in the simulation tests section, whereas the physical load is a magnetic load [25], as it is shown in Fig. 1, and in order to reach the 200 RPM speed reduction equivalent load, the operator should start the movement of the load

Fig. 16 Performance indices practical values in graph form

Table 3 Performance indices practical results

Performance index Real-time values	Cross-coupling		Master–Slave	
	PMDC Motor 1	PMDC Motor 2	PMDC Motor 1	PMDC Motor 2
IAE	1.5917	1.5826	1.5435	1.8424
ISE	0.7848	0.7757	0.7838	1.1679
ITAE	1.79	1.88	1.68	2.32
ITSE	0.9662	0.9543	0.9311	1.2929

mechanism. This consumes more time and implements different loads on the motor shaft before the intended load is reached. So, the physical performance indices have larger values than the ideal ones.

6 Conclusion

In this paper, a comparison based on the performance indices is made between two popular strategies for speed synchronization. Those two methods are cross-coupling method and the master–slave method. Two identical PMDC motors are used. The system identification procedure is carried out using a pseudo-random binary sequence (PRBS) test and the transfer function is derived. Optimal tuning of the Proportional-Integral (PI) controller is performed based on the multiple-application Simpson's 1/3 rule with the performance indices as objective functions. Both cross-coupling and master–slave speed synchronization strategies are implemented. A comparison is made for their performance using the performance indices. A microprocessor-based embedded system, i.e. ARDUINO Mega board along with MATLAB SIMULINK (after installation of MATLAB support package for ARDUINO) are used to implement the developed controller for speed synchronization between two PMDC motors and to make the mentioned comparison. Matching between simulation results and practical results are obtained.

The time delay taken by the slave motor to follow the master motor makes its performance indices having larger values. The performance indices simulation obtained values are smaller or ideal values in comparison with the performance indices physical obtained values due to the consideration of the estimated mathematical models of the utilized actuators in the simulation tests. The current work is a step toward further development in the distributed motion control domain. The recent direction in the industry is to use the state of the art technologies for remote control and observations, for example, the internet of things technology (IoT) integrates with the controllers to result in industrial IoT (IIoT), so current work is a pre-IIoT integration process and it is resulted in selection of a strategy with optimal controller behavior at edge level for IIoT integration procedure. In addition to that, other strategies are to be tested and merged, as a future scope of current work, in order to obtain a strategy with best controller performance and integrate it with IIoT technology. For the academic world, current work enables the students to find a general methodology that teach them how to use the low cost and readily made available component starting from basic level in the motion control related applications.

Acknowledgements The corresponding author would like to thank Prof. V. P. Patel, I&C Department, L D College of Engineering, Ahmadabad, India, for the valuable information during the corresponding author M.E. course there.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Paresh CS (1990) Electric motor drives and control past, present and future. *IEEE Trans Ind Electron* 37(6):562–575
2. Okan B, Serdar P (2012) A virtual laboratory for neural network controlled DC motors based on a DC–DC buck converter. *Int J Eng Educ* 28(3):713–723
3. Okan B, Serdar P (2010) A virtual laboratory for fuzzy logic controlled DC motors. *Int J Phys Sci* 5(16):2493–2502
4. Rahman M, Hoque M (1997) On-line self-tuning ANN-based speed control of a PM DC motor. *IEEE/ASME Trans Mechatron* 2(3):169–178
5. Zhang D, Chen Y, Tom KC et al (2007) Compensation scheme of position angle errors of permanent-magnet linear motors. *IEEE Trans Magn* 43:3868–3871
6. Tan KK, Lee TH, Dou HF, Chin SJ, Zhao S (2003) Precision motion control with disturbance observer for pulse-width-modulated driven permanent-magnet linear motors. *IEEE Trans Magn* 3:1813–1818
7. Otten G, de Vries TJA, van Amerongen J, Rankers AM, Gaal EW (1997) Linear motor motion control using a learning feed-forward controller. *IEEE/ASME Trans Mechatron* 2:179–187
8. Yeh SS, Hsu PL (2003) Analysis and design of integrated control for multi-axis motion systems. *IEEE Trans Control Syst Technol* 11:375–382
9. Kothari VK (2013) Process control in weaving. In: *Process control in textile manufacturing*, pp 265–278. <https://doi.org/10.1533/9780857095633.3.265>
10. Perez-Diaz F, Trenkwalder SM, Zillmer R, Groß R (2018) Emergence and inhibition of synchronization in robot swarms. In: Groß R et al (eds) *Distributed autonomous robotic systems*. Springer Proceedings in Advanced Robotics, vol 6. Springer, Cham
11. Kulkarni PK, Srinivasan K (1990) Cross-coupled control of biaxial feed drive servomechanisms. *ASME J Dyn Syst Meas Control* 112(2):225–232
12. Moore PR, Chen CM (1995) Fuzzy logic coupling and synchronized control of multiple independent servo-drives. *Control Eng Pract* 3(12):1697–1708
13. Lee HC, Jeon GJ (1998) A neuro-controller for synchronization of two motion axes. *Int J Intell Syst* 13:571–586
14. Sun D, Mills JK (2002) Adaptive synchronized control for coordination of multi-robot assembly tasks. *IEEE Trans Robot Autom* 18:498–510
15. Xiao Y, Zhu K (2006) Optimal synchronization control of high-precision motion systems. *IEEE Trans Ind Electron* 53:1160–1169
16. Perez-Pinal FJ, Nunez C, Alvarez R, Cervantes I (2004) Comparison of multi-motor synchronization techniques. In: *Proceedings of the 30th annual conference of the IEEE Industrial Electronics Society* (2004)
17. Fing L, Koren Y, Borenstien J (1993) Cross-coupling motion controller for mobile robots. *IEEE* 13:35–43

18. Tomasuka M, Kamona T, Hu JS, Chiu TC (1992) Synchronization of two motion control axes under adaptive feedforward control. *J Dyn Syst Meas Control* 114:196–203
19. Yeh SS, Hsu PL (1997) Theory and applications of the robust cross-coupled control design. In: American Conference
20. Xiao Y, Pang Y, Ge X, Sun J (2013) Synchronous control for high-accuracy bi_axial motion system. *J Control Theory Appl* 11:294–298
21. Zeroug H, Boudjit K, Kachroud H, Sahraoui H (2010) Speed synchronization control of multiple DC motors using DSP. In: Proceedings of the 4th European DSP in Education and Research Conference
22. Krüger J, Wang L, Verl A, Bauernhansl T, Carpanzano E, Makris S, Fleischer J, Reinhart G, Franke J, Pellegrinelli S (2017) Innovative control of assembly systems and lines. *CIRP Ann Manuf Technol* 66:707–730
23. Spiliotopoulos J, Michalos G, Makris S (2018) A reconfigurable gripper for dexterous manipulation in flexible assembly. *Inventions* 3(1):4. <https://doi.org/10.3390/inventions3010004>
24. Soliman WG, Reddy DVRK (2019) Microprocessor-based edge computing for an Internet of Things (IoT) enabled distributed motion control. In: Singh M, Gupta P, Tyagi V, Flusser J, Ören T, Kashyap R (eds) *Advances in computing and data sciences. ICACDS 2019. Communications in Computer and Information Science*, vol 1046. Springer, Singapore. https://doi.org/10.1007/978-981-13-9942-8_3
25. Ghder Soliman W, Rama Koti Reddy DV (2019) Microprocessor-based prototype design of a PMDC motor with its system identification and PI controller design. *SN Appl Sci* 1:549. <https://doi.org/10.1007/s42452-019-0579-0>
26. Wasim S et al (2018) Microprocessor-based permanent magnetic DC motor system identification and optimal PI controller design. In: Proceedings of the 12th INDIACom, IEEE Conference ID: 42835, pp 1020–1026
27. Martins FG (2005) Tuning PID controllers using the ITAE criterion. *Int J Eng Educ* 21:867–873
28. Chapra SC, Canale RP (2015) *Numerical methods for engineers*, 7th edn. Wiley, New York

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.