

Article

A Sensorless and Low-Gain Brushless DC Motor Controller Using a Simplified Dynamic Force Compensator for Robot Arm Application

Shih-Hsiang Yen ¹, Pei-Chong Tang ², Yuan-Chiu Lin ² and Chyi-Yeu Lin ^{1,3,4,*} 

¹ Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan

² Ubiquelife Technology Corporation, Jhubei City, Hsinchu 302, Taiwan

³ Taiwan Building Technology Center, National Taiwan University of Science and Technology, Taipei 106, Taiwan

⁴ Center for Cyber-Physical System, National Taiwan University of Science and Technology, Taipei 106, Taiwan

* Correspondence: jerrylin@mail.ntust.edu.tw

Received: 14 June 2019; Accepted: 16 July 2019; Published: 18 July 2019



Abstract: Robot arms used for service applications require safe human–machine interactions; therefore, the control gain of such robot arms must be minimized to limit the force output during operation, which slows the response of the control system. To improve cost efficiency, low-resolution sensors can be used to reduce cost because the robot arms do not require high precision of position sensing. However, low-resolution sensors slow the response of closed-loop control systems, leading to low accuracy. Focusing on safety and cost reduction, this study proposed a low-gain, sensorless Brushless DC motor control architecture, which performed position and torque control using only Hall-effect sensors and a current sensor. Low-pass filters were added in servo controllers to solve the sensing problems of undersampling and noise. To improve the control system’s excessively slow response, we added a dynamic force compensator in the current controllers, simplified the system model, and conducted tuning experiments to expedite the calculation of dynamic force. These approaches achieved real-time current compensation, and accelerated control response and accuracy. Finally, a seven-axis robot arm was used in our experiments and analyses to verify the effectiveness of the simplified dynamic force compensators. Specifically, these experiments examined whether the sensorless drivers and compensators could achieve the required response and accuracy while reducing the control system’s cost.

Keywords: sensorless; Hall-effect sensors; dynamic compensator; low-gain control; low-pass filter; real-time control

1. Introduction

Industrial robots have been developed for decades and have served as a solution for the two automation insufficiencies: Quality uniformity and labor force. The number of robot arms in service applications is expected to exceed those in industrial applications by 2020 [1]. However, large-scale robot arm applications have not been observed in the service domain because their characteristics of quickness and high precision are accompanied by drawbacks of heaviness, low payload-to-weight ratio, and high risk. Robot arms used for service applications (hereafter referred to as service robots) do not require the same specifications as industrial robots; they instead emphasize mobility and safety of human–robot interaction [2]. In 2006, KUKA (Germany) launched the first lightweight robot arm, (LWR) [3]; the LWR’s elastic joint design [4], which incorporates torque sensors and provides safe control, has influenced various subsequent collaborative robot designs, such as the UR Robot

(Universal Robots) and YuMi (ABB). However, despite the effective and safe control provided by the LWR design, its unique structure combining elastic joints and numerous sensors leads to high production cost, thus failing to achieve a favorable cost–performance ratio and limiting its service applications. A service robot does not require high-precision position control; thus, fewer sensors are required for service robots than for industrial robots. For example, replacing the encoders of position sensors with Hall-effect sensors can reduce the amount of additional sensor installation required; therefore, such robot arms exhibit lower production costs and wider potential applications in service domains.

Studies related to sensorless control of Brushless Direct Current (BLDC) motors have used Hall-effect sensors to perform motor position and speed control, although the low-resolution Hall sensors may result in signal undersampling. The majority of research, shown in Table 1, proposed the incorporation of low-pass filters in control as a solution for undersampling and noise but resulted in signal phase delay problems. The signal delay may be ignored or not apparent in the high-speed application, such as in [5]. In contrast, the delay issue in the low-speed application significantly reduced control accuracy. The compensator was an excellent way to solve the issue of signal delay. In Refs. [6–9], they designed compensator based on system models and experiments to achieve an excellent control effect in the simple or no-load system. However, this sensorless control study has not been applied in robot arm application, because their inherent problems of signal delay and low response prohibit the real-time control required by complicated dynamic robot systems.

Table 1. Comparison of sensorless Brushless Direct Current (BLDC) motor control research.

References	[6]	[7]	[8]	[9]	Proposed Approach
Speed range (rpm)	19–4600	20–1000	1000–4800	200–3000	100–1000
Filter	Low-pass filter	Low-pass filter	Low-pass filter	Low-pass filter	Low-pass filter
Motor poles	4	4	4	4	4
Application	No load	Petroleum drilling system	Driveline system	No load	Robot arm
Solution of phase-delay	Flux linkage threshold	Adaptive compensation	Active compensation	Novel speed calculation	Dynamic compensation

Service robots share workspaces with their users; therefore, for safety at work, such robots should have limited velocity (<250 mm/s) and control force in robot motion [10]. Low-gain control is a safe control approach that avoids producing excessive unspecified forces, but low-gain control exhibits insufficient torque, which can lead to slow control response and low dynamic accuracy. Given a system model, the open-loop control approach can provide an excellent control response [11] and directly control motors in advance without any sensors providing status feedback. For example, current feedforward controllers can enhance the response of dynamic control. In Refs [12,13], they have incorporated robot dynamic equations into a robot controller to get better performance. However, these researches require a highly accurate system model, and the complex component configuration of robots yields great difficulty in obtaining the numerous system parameters, such as the moment of inertia and friction, etc. Moreover, the calculation of the dynamic equation involves complex multidimensional matrix operations, which requires long coding times and high costs, making it mostly infeasible on real-time robot control in reality.

Focusing on increasing control safety and reducing cost, we proposed a low-gain control architecture equipped with sensorless drivers and dynamic force compensators. In our previous research [14], we have presented a safety control method using dynamic compensators to detect an external force. In this study, the proposed architecture is intended to solve the problems of delayed signals in servo control and insufficient torque in low-gain control as well as to increase the dynamic control response and position accuracy. To address the complexity of system models, we employed a method to simplify dynamic equations and developed a controller to realize a seven-axis robot arm.

Our experimental results proved the accuracy of the sensorless position control and confirmed that this control architecture achieves the real-time motion control effectiveness required of service robots.

2. Sensorless Control Problems

This study used sensorless drivers to control BLDC motors for low-cost robot application. Only Hall-effect sensors and a current sensor are used to perform position and current control. We have designed three joint modules with BLDC motors and harmonic drives according to the required torque (specifications presented in Table 2), with a minimum joint position resolution of smaller than 0.2° . To reduce the size of driver, this study used embedded drivers designed from a development board of Field-Programmable Gate Array (FPGA). The driver dimension is $80 \times 65 \times 35 \text{ mm}^3$ with a double-layer power circuit board shown in Figure 1. Each driver controls two BLDC motors.

Table 2. Specification of 3-phase BLDC motors.

Parameter	Type 1	Type 2	Type 3
Joint	J1–J4	J5/J7	J6
Weight	1050 g	150 g	150 g
Voltage	24 V	24 V	24 V
Rated speed	1000 rpm	4840 rpm	4840 rpm
Rated current	3.6 A	3.26 A	3.26 A
Rated torque	0.6 N·m	0.13 N·m	0.13 N·m
Torque constant	0.16 N·m/A	0.037 N·m/A	0.037 N·m/A
Motor poles	4 pairs	8 pairs	8 pairs
Resolution	24 pulse/rev	48 pulse/rev	48 pulse/rev
Gear rate	1:80	1:50	1:120
Min. joint resolution	0.1875°	0.15°	0.0625°
Max. joint speed	$75^\circ/\text{s}$	$576^\circ/\text{s}$	$242^\circ/\text{s}$

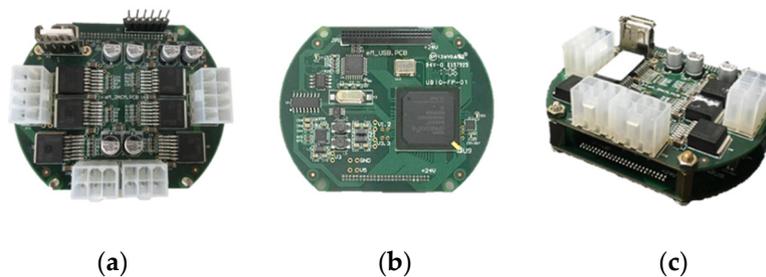


Figure 1. Embedded Field-Programmable Gate Array (FPGA)/Digital Signal Processor (DSP) driver: (a) layer1; (b) layer2; (c) assembled driver.

Hall sensors used in low-speed operation cause an undersampling problem, which results in discontinuous and high-frequency oscillation; thus, they are unsuitable for highly adaptive Proportional-Integral-Derivative (PID) controllers. Therefore, we propose in this chapter a low-pass filter capable of velocity adaptation to solve this problem.

2.1. Low-Speed Control with Low-Pass Filter

Sensorless current control was performed using a single-phase current sensing approach [15], which involved the use of only one current sensor and measuring phase Back Electromotive Force (EMF) to predict the current of the BLDC motors. However, this approach measured only the magnitude of current and could not detect current direction. To solve this problem, we adopted reference values based on output voltage directions, but this approach also exhibited a problem; when a current approached the zero-crossing point, the voltage led to high-frequency phase shifts, which resulted in incorrect current detection.

To obtain the motor velocity in sensorless control, the position feedback information must be changed for each sample. With an assumption that the motor's control resolution is n [pulse/rev], the sampling frequency is f_s [Hz], the minimum detectable velocity v [rpm] is given by (1). After substitution of detectable velocity range, which is 100–1000 rpm, and the Type 1 motor's (Table 1) resolution into the equation, the sampling frequency must be 40–400 Hz.

$$v = \frac{f_s}{n} \times 60 \quad (1)$$

With the position resolution constant, this study changed the sampling frequency to solve these problems of speed and current detection [9]; we designed a Low-Pass Filter (LPF), which adjusted the sampling frequency f_s according to the velocity command v , as presented in (2) where k_T is sampling gain. The LPF could adapt to a wide range of velocity control and stabilize velocity signals by adjusting sampling frequency, also it can filter out high-frequency noises or oscillations from current signals.

$$f_s = k_T |v| \quad (2)$$

This study incorporated three LPFs into the current controller and velocity controller to solve the signal discontinuity and high-frequency noise. The controller architecture on FPGA driver is shown in Figure 2. In current control, the LPF filtered out high-frequency noises and oscillations. Figure 3 compares the voltage signal outputs with and without the LPF. In velocity control, one LPF filtered the discontinuous signal on the velocity feedback and the other LPF smoothed the output signal on the velocity controller. Figure 4 compares the outputs with and without the LPF, thus producing more stable and comprehensive output signals in low-speed operation.

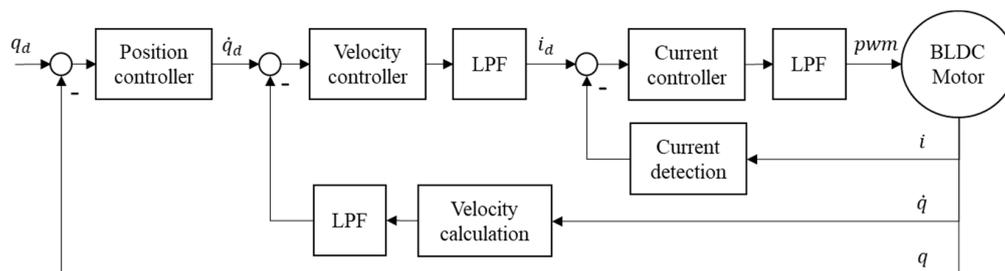


Figure 2. Servo control diagram on FPGA driver.

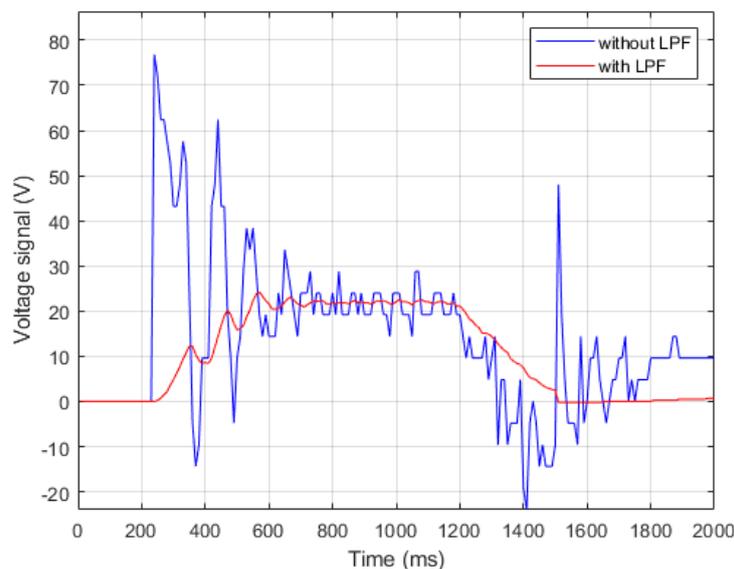


Figure 3. Voltage signal output on current controller.

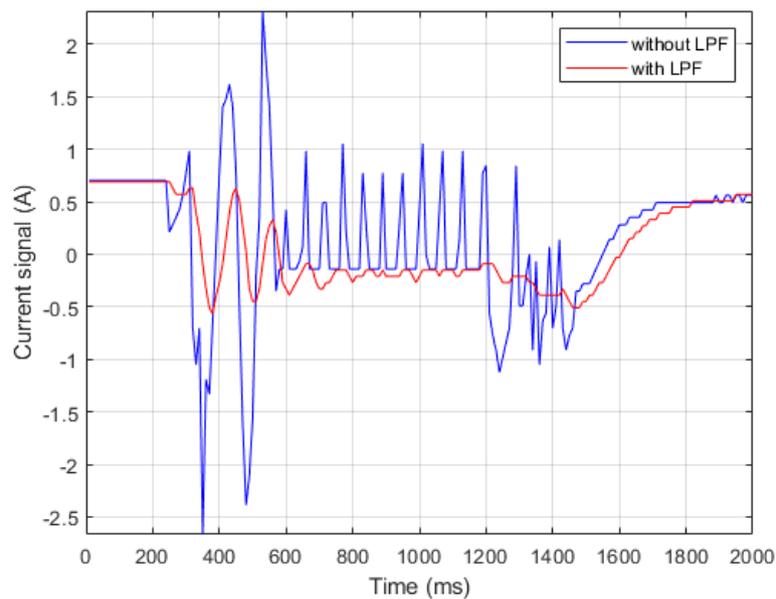


Figure 4. Current signal output on velocity controller.

2.2. Phase Delay Problem

We added three LPFs in the closed-loop controllers to produce smooth and continuous output signals. However, these filters increase the sampling period and thus caused signal phase delay. Based on our experiment, the current output exhibited a phase delay of approximately 50–500 ms under different velocities. Such delay slowed the response of the closed-loop control and reduced the accuracy of dynamic control.

To accelerate the controller's response, we added a velocity feedforward controller, which provided compensation through velocity commands, thus reducing the controller delay caused by the LPFs. Since the operation of the current feedforward controller required force information obtainable only from the robot arm motion, the current feedforward controller could not operate separately in each single-axis driver.

3. Simplified Robot Dynamic Compensator

Robot arms are nonlinear systems whose parameters are highly complex and difficult to obtain. Therefore, dynamic force calculation in a program is difficult and entails high costs and long calculation times. This chapter discusses the approaches we used to reduce development cost, namely simplifying the dynamic models and reducing the number of parameters, and how we increased the accuracy of the model parameters by experiments.

3.1. Robot Dynamic Model

The ideal dynamic model of an n -joint manipulator can be written in the Lagrangian form as [16]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3)$$

where $q \in \mathbb{R}^{n \times 1}$ is the joint variable vector [rad], $\tau \in \mathbb{R}^{n \times 1}$ is the vector of generalized torque [N·m], $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix [kg·m²], $C(q, \dot{q})\dot{q} \in \mathbb{R}^{n \times 1}$ is the vector of Coriolis and centripetal torque [N·m], and $G(q) \in \mathbb{R}^{n \times 1}$ is the vector of gravity torque [N·m]. Under single-joint motor steady-state operation [17], electromagnetic torque will be counter-balanced by load torque, inertia torque, and friction torque as Figure 5. Therefore,

$$M_m \ddot{\theta}_m + B_m \dot{\theta}_m = \tau_e - \frac{\tau_l}{r} \quad (4)$$

$$q = \theta_s = \frac{\theta_m}{r} \quad (5)$$

where τ_e is the electromagnetic torque, τ_l is the load torque [N·m], M_m is the inertia constant of the rotor and coupled shaft [kg·m²], B_m is the friction constant factor [N·m·s/rad], θ_m is the motor rotation angle [rad], θ_s is the joint rotation angle [rad], and r is the gear ratio.

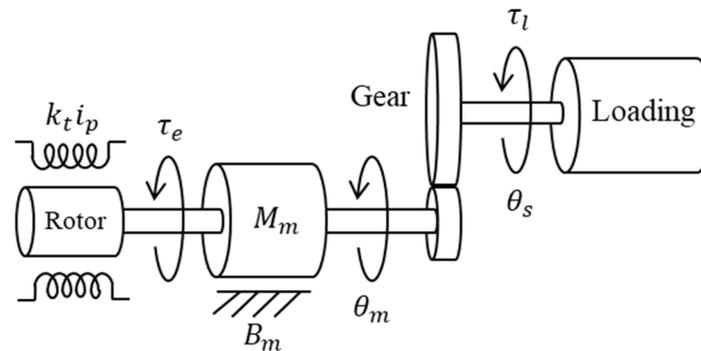


Figure 5. Lumped model of a single link with actuator/gear train.

Using (4) and (5), which expands calculation from a single-joint to n-joint, into (3), we obtain the motor dynamic Equation (6), where $M(q)$, $C(q, \dot{q})$, and $G(q)$ are related to posture and motion of the robot arm, and $M_m \in \mathbb{R}^{n \times n}$ and $B_m \in \mathbb{R}^{n \times n}$ are the constant matrices of motors. In the constant loading application, the dynamic inertia, Coriolis/centripetal, and gravity torque can be ignored. However, the serial robot arm has drastic changes in dynamic inertia and gravity torque with different postures. Therefore, the electromagnetic torque must be real-time calculated from each joint position feedback and robot arm model.

$$\tau_e = \left(M_m + \frac{M(q)}{r^2} \right) \ddot{\theta}_m + \left(B_m + \frac{C(q, \dot{q})}{r^2} \right) \dot{\theta}_m + \frac{G(q)}{r} \quad (6)$$

3.2. Simplified Robot Dynamic Model

The design of robot arms entails many parts of various materials and shapes, which reduce the accuracy of system model estimation [18], and requires numerous system parameters and a complex process to calculate dynamic force. To reduce the difficulty of obtaining and calculating system parameters, this study simplified a robot arm's continuous mass model into mass points system according to its link system (Figure 6). To account for inertia and gravity torque calculation, this study adopted the Universal Robot Description Format (URDF) [19] to define the robot arms' coordinate system (Table 3). To simplify relevant definitions in the URDF model, the coordinate directions of each joint are set to be the same directions as the world coordinate system. In Table 3, X, Y, and Z represent the offset of each joint coordinate, M refers to the mass of the link, CM denotes the offset of a link's center of mass from the joint coordinate, and R gives the rotational directions of the motors. This study used a robot arm under a no-loading condition; the mass of Link 7 and center of mass could be adjusted if grippers or any other components were installed on the end-effector.

To calculate the moment of inertia, this study simplified the parallel axis theorem according to mass points system as shown in Figure 7, where I_P is the inertia of point P, I_{CM} is the default inertia based on the center of mass, such as that of a motor rotor and all structures, m is mass, and d is the offset between the link's center of mass and point P. This moment of inertia was difficult to obtain through models; therefore, we conducted experiments to obtain these moments of inertia.

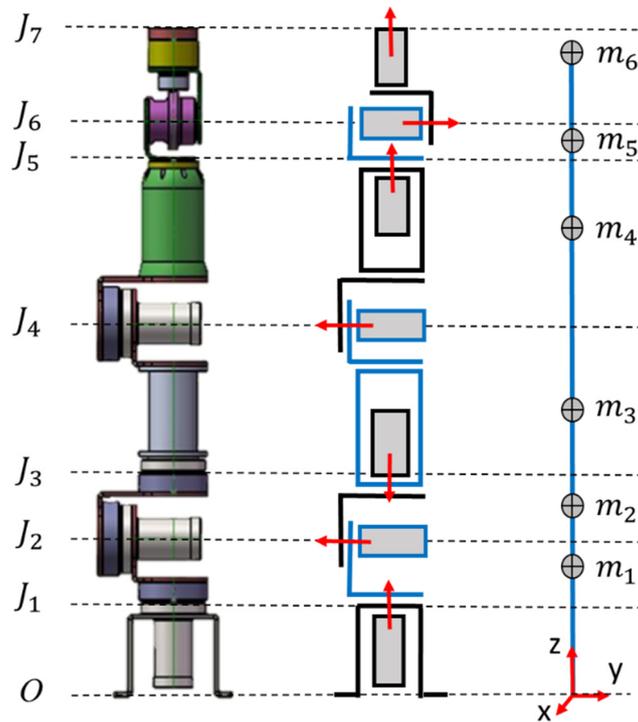


Figure 6. Simplified mass-point model on a seven-axis robot.

Table 3. Universal Robot Description Format (URDF) table of a seven-axis robot arm.

Joint	X [mm]	Y [mm]	Z [mm]	M [kg]	CM [mm]	R [axis]
J1	0.0	0.0	156.0	2.95	31.0	Z
J2	0.0	0.0	58.5	2.17	43.7	-Y
J3	0.0	0.0	63.5	4.35	147.7	-Z
J4	0.0	0.0	229.5	2.43	86.6	-Y
J5	0.0	0.0	222.4	1.24	48.5	Z
J6	15.0	0.0	53.0	1.07	70.5	Y
J7	0.0	0.0	110.7	0.0	0.0	Z

CM = Center of Mass.

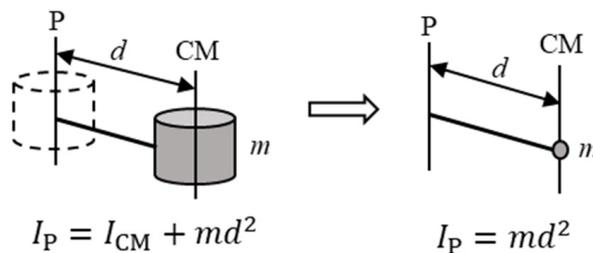


Figure 7. Simplification of the parallel axis theorem.

Dynamic inertia changes with the center of mass offsets; only the inertia component on the motor from offset (d_{ki}) is considered, which was projected from \vec{r}_{ki} onto the motors' rotation plane. Figure 8 shows an example for inertia calculation on Joint 1. Calculation of the two elements' inertia was simplified as $k_{a0} + k_a M_k$, where k_{a0} is a constant inertia coefficient, k_a is a dynamic inertia coefficient, and M_k is the dynamic inertia of k^{th} joint, as presented in (7).

$$M_k = \sum_{i=k}^7 m_i \cdot d_{ki}^2 = \sum_{i=k}^7 m_i \cdot \left| \vec{r}_{ki} \times \vec{\varepsilon}_k \right|^2 \tag{7}$$

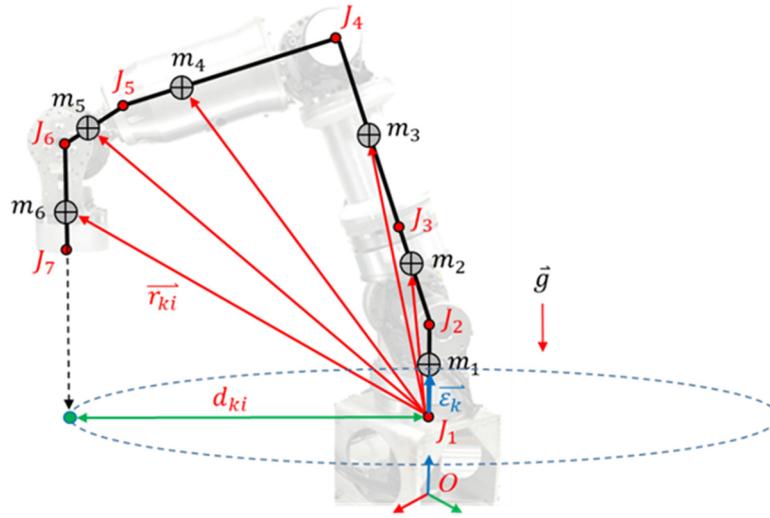


Figure 8. Dynamic inertia calculation of k th joint (example for $k = 1, i = 6$).

Regarding the calculation of gravity torque, we calculate only the torque based on each link's center of mass projection on the motors' rotation direction. Specifically, gravity torque was simplified as $k_g G_k$, where k_g is a gravity coefficient and G_k is the gravity torque of k th joint. In Equations (7) and (8), \vec{r}_{ki} refers to the vector from k th joint to i th link's center of mass, m_i is the mass of i th link, \vec{g} denotes gravitational acceleration, and $\vec{\varepsilon}_k$ is the unit vector of k th joint's rotating direction.

$$G_k = \sum_{i=k}^7 (\vec{r}_{ki} \times m_i \vec{g}) \cdot \vec{\varepsilon}_k \quad (8)$$

According to Equation (9), Coriolis and centripetal torques are related to the partial differentials of inertia [20]. In this study, the robot arm's lightweight structural design produced relatively small inertia in its links, thus minimizing the influence on the arm's overall motion. To favor calculation efficiency, this study disregarded the compensation for Coriolis and centripetal torques.

$$C_{ij}(q, \dot{q}) = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i} \right) \dot{\theta}_k \quad (9)$$

A friction identification experiment conducted by Wolf et al. [21] revealed that the friction in joint mechanisms was a complex nonlinear system; therefore, considering calculation efficiency, we employed a simplified friction Equation (10), where \dot{q} is the motor angular velocity, k_{v0} and k_v are the coefficients of Coulomb friction and viscous friction, respectively. These two coefficients can be obtained through experiments.

$$B_m = k_{v0} \text{sgn}(\dot{q}) + k_v \dot{q} \quad (10)$$

Using the simplified process, we further simplified (6) as a new motor dynamic Equation (11), in which only dynamic inertia (M_k) and gravity (G_k) require kinematic model calculation; other simplified coefficients could be obtained through experiments as shown in Table 4. Accordingly, our simplification approach reduced calculation process and improved the efficiency of program execution.

$$\tau_d = (k_{a0} + k_a M_k) \ddot{q} + k_{v0} \text{sgn}(\dot{q}) + k_v \dot{q} + k_g G_k \quad (11)$$

Table 4. Coefficient definition on the dynamic equation.

Physical	Coefficient
Default moment of inertia	k_a
Dynamic moment of inertia	k_{a0}
Viscous friction	k_v
Coulomb friction	k_{v0}
Gravity torque	k_g

3.3. Simplified Dynamic Force Compensator Design

In Equation (11), state information is represented by motor position (q), velocity (\dot{q}), and acceleration (\ddot{q}) and is usually obtained through the following two approaches.

- Control commands: These calculate trajectories offline in advance and generate continuous and smooth signals [12,22]; however, they cannot reflect the real state of systems when excessively large errors occur.
- Sensor feedback: This facilitates obtaining real-time state information but requires additional high-resolution sensors, such as accelerometers [23] and high-speed communication systems. This study employed only low-resolution position sensors, and thus approach (b) was not feasible. Moreover, in approach (a), the low-gain controller possessed large errors on differences between sensor feedback and command. To solve these problems, we proposed a hybrid approach (c).
- Hybrid approach: A trajectory generator's control commands are used to calculate velocity and acceleration. Position states are feedback through sensors to facilitate real-time position updates for each motor.

We substituted state information obtained through approach (c) into (11) and obtained (12), which yielded a Simplified Dynamic Force Compensator (SDFC), as presented in Figure 9 [13]. In this compensator, the calculations of dynamic inertia and gravity torque (M_k and G_k) by forward kinematics, both of which require n-joint position information, are conducted separately in the robot controller with a period of 20 ms as the green block in Figure 9. Other compensation calculations are conducted in a FPGA driver operating with 1-ms interrupts. Under such a control system, the servo controller's torque output, τ_c , could be considered a, which is the sum of the external disturbance torque and the error of the compensator's computed torques.

$$\tau_d = (k_{a0} + k_a M_k) \ddot{q}_d + k_{v0} \text{sign}(\dot{q}_d) + k_v \dot{q}_d + k_g G_k \quad (12)$$

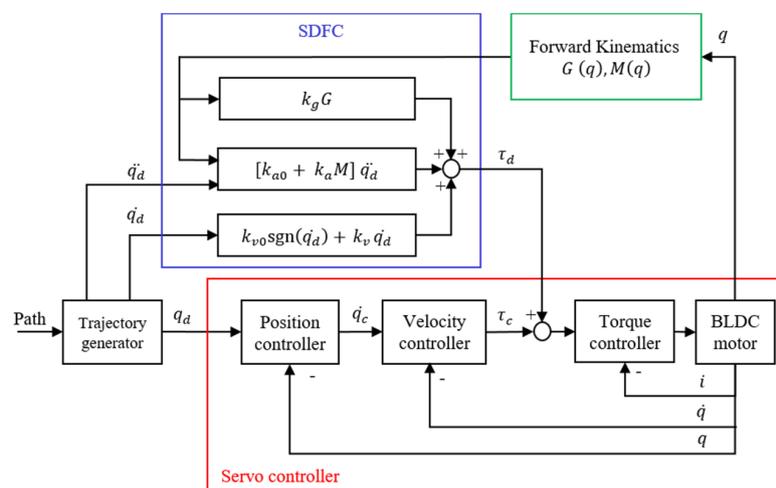


Figure 9. Simplified Dynamic Force Compensator (SDFC) control diagram.

4. Simplified Dynamic Force Compensator Tuning

In the SDFC design, we simplified system models by reducing the number of system parameters; specifically, the robot controller calculates only the dynamic inertia and gravity torque by robot posture model, and the FPGA driver facilitates adjustments in compensation gain through experiments, which generate results more applicable to actual systems. The following sections discuss the experimental design used to adjust the gain parameters in the SDFC.

4.1. Simplifying the Calculations of Moment of Inertia and Gravity Torque

Equations (7) and (8) expedited our calculations of dynamic inertia and gravity torque (M_k and G_k), but the resulting floating-point number was excessively large for real-time communication and could not be efficiently processed in the integral calculation of DSP in the FPGA driver. Therefore, we normalized the calculated inertia and gravity, namely multiplying the calculated results by proportional coefficients (P_M and P_G) to transform these results into integers from -100 to 100 . Taking the seven-axis robot arm as an example, the maximum result of each joint and proportional coefficients are presented in Table 5. With no-load on end-effector, there was no dynamic inertia and gravity torque on Joint 7. Proportional errors resulting from normalization could be reduced by adjusting the gains (k_a and k_g) through experiments. Finally, the dynamic inertia and gravity torque are included in real-time control commands, such as position and velocity command, between the robot controller and FPGA drivers for updating real-time robot status.

Table 5. Maximum dynamic inertia and gravity torque.

Joint	M_k [kg·m ²]	P_M	G_k [kg·m]	P_G
J1	1.3790	72.52	0	0
J2	1.3795	72.50	3.3165	30.15
J3	0.2377	420.70	0.9175	108.99
J4	0.2382	419.64	0.9175	108.99
J5	0.0081	12,345.68	0.1090	917.43
J6	0.0054	18,518.52	0.0758	1,319.26

4.2. Tuning Experiment with SDFC

To adjust the compensator's gain, this study designed three experiments to measure the currents when dynamic forces occur. We analyzed gravity, friction, and inertia torques separately for each joint and adjusted the gain to generate a compensatory force that approximates the actual force of real robot systems. The experimental steps for each joint were as follows:

- Gravity compensator tuning: the force resulting from gravity acting upon the robot arm changes only with its posture; therefore, we first adjusted the gravity compensator. In the experiment, we adjusted the robot joints to a posture that yielded the greatest gravity torque and obtained a computed gravity parameter G_k of 100. A stable motor current i_g under this posture was measured and substituted into (13), which yielded a gravity gain k_g .

$$i_g = k_g G_k \quad (13)$$

- Friction compensator tuning: after adjusting the gravity compensator, a gravity torque can be disregarded in control processes; thus, the next step is friction compensator adjustment. Because friction only occurs during the motion of motors and is positively related to velocity, we conducted controlled experiments with constant velocity to measure servo currents and obtained average currents i_f under different velocities as the blue dots in Figure 10. Subsequently, the results were used to conduct a linear regression analysis, which yielded two linear equations, namely the red lines in Figure 10. Using (14), the two friction coefficients k_v and k_{v0} could be obtained from the

regression line as the slope and the y-intercept, respectively. Finally, the gain coefficients of the two directions were averaged and incorporated into this step, and this process was repeated for further adjustment.

$$i_f = [k_{v0} \operatorname{sgn}(\dot{q}) + k_v \dot{q}] \quad (14)$$

- (c) Inertia compensator tuning: inertia force occurs only during the acceleration and deceleration stages; because these two stages usually have short durations (<1 s) and suffer from current signal delay, accurate current values are rarely obtained. Therefore, this study used a numerical approximation to adjust inertia gains (k_a and k_{a0}) rather than using currents to predict them. First, we generated a trajectory for a given acceleration. Under low-inertia conditions (mainly default inertia), k_{a0} was adjusted to reduce the errors of servo currents in the acceleration and deceleration intervals; under high-inertia conditions (mainly dynamic inertia), k_a was adjusted to the same step as k_{a0} adjustment. These two steps facilitated the inertia calculation in (15) to more accurately approximate real systems.

$$i_{\text{inertia}} = [k_{a0} + k_a M_k] \ddot{q} \quad (15)$$

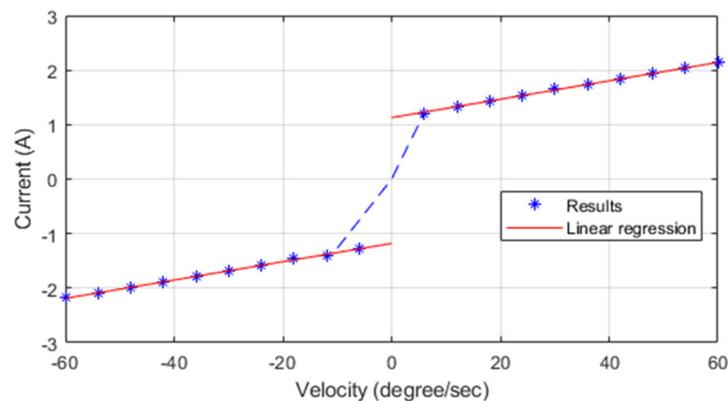


Figure 10. Friction identification experiment.

Comprehensive force compensation was achieved after the three aforementioned adjustment steps. For example, the position control results in Joint 2 is illustrated in Figure 11, in which the red line represents SDFC's output current, the blue line represents the servo controller's output current, and the black line represents the sum of these two's output currents. Because the SDFC provided the needed currents for motion, the servo controller was required only to reduce position errors; this greatly reduced servo controller current output. The tuning experiments yielded the compensation gains for the seven-axis SDFC; these are listed in Table 6.

Table 6. SFDC compensation gain.

Joint	k_g	k_v	k_{v0}	k_a	k_{a0}
J1	0	40	145	200	50
J2	-1250	40	180	150	100
J3	-230	50	220	150	67
J4	-230	50	200	100	100
J5	-120	10	100	10	20
J6	-50	35	80	40	50
J7	0	30	160	0	30

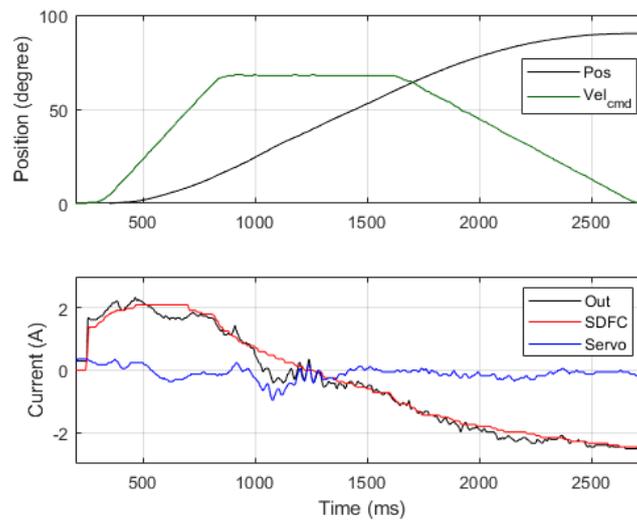


Figure 11. Current output performance for Joint 2 with an SDFC.

4.3. Analysis of Phase Compensation

Before the incorporation of the SDFC, the phase delay in the control signals caused by the low-gain controller and LPFs resulted in low-accuracy position control. For example, Figure 12 presents the result of Joint 1's position control (velocity of $60^\circ/\text{s}$) and reveals that the closed-loop servo control without SDFC leads to a delay of 50 ms and position error greater than 10° (blue line). After incorporating the SDFC, the force compensator reduced the dynamic error to less than 1° (red line). Comparing the current output results with and without the SDFC, current response was faster with compensation; this verified that the SDFC increased control response and reduced dynamic error.

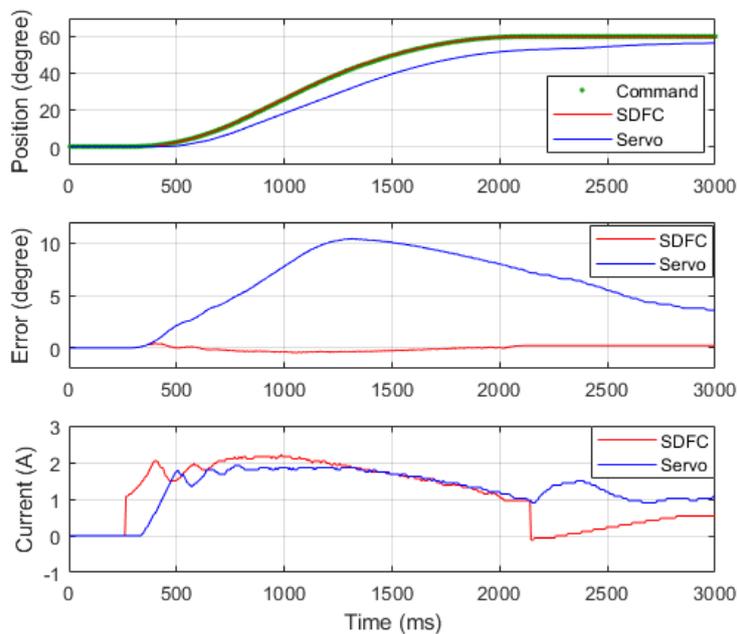


Figure 12. Results comparison with SDFC and closed-loop servo controller.

5. Real-Time System Framework

To achieve an efficient control effect and dynamic compensations, a robot controller must transmit control information to each motor driver in real-time. The control frequency is usually set higher than 1 kHz, and the controllers must be equipped with a Real-Time Operating System (RTOS) and real-time communication to avoid system delays, which could cause discontinuity and further delays.

This study used open-source software to design a real-time robot controller prioritizing low cost and employed the Raspberry Pi 3 Module B (Raspberry Pi Foundation, Cambridge, UK) and Linux Ubuntu MATE system accompanied by a Linux Xenomai 3 RTOS to establish a real-time platform. This chapter explains the real-time architecture of our controller.

5.1. RTOS–Xenomai

Xenomai [24] is an open-source RTOS provided on the Linux platform, uses a dual kernel structure and Adaptive Domain Environment for Operating Systems (ADEOS) to manage cokernel and user-kernel boundaries, which have high priority, and schedules real-time and non-real-time tasks in CPU processing. While other RTOSs [25] focused on the lowest technically feasible latencies, Xenomai also considers clean extensibility (RTOS skins), portability, and maintainability as very important goals.

Compared with a previous version, Xenomai 2.6, Xenomai 3 has improved performance of the Cobalt kernel, which reduces latency to 10 μs on a CPU-stressed setup. It provides more skins and classifies different application programming interface libraries according to their applications; it enables users to select different skins according to their needs and simplify and expedite program design. Regarding this study's real-time controller, we used the Xenomai's Native skin to design a real-time motion control program. To facilitate real-time communication, we used the Real-time Drive Model (RTDM) skin [26] to design a real-time communication architecture.

5.2. Real-Time Communication

A standard CANopen system (an industrial control network) [27] is equipped with two real-time communications, namely CANbus and EtherCAT. In particular, CANbus exhibits low cost and high anti-interference but has a short data length of 16 bytes; therefore, it is not suitable for real-time multi-node control. EtherCAT demonstrates the advantage of high information throughput, indicating its suitability for real-time multi-node control, but is costly in its hardware and communication chips.

Considering the aforementioned factors and attempting to build controllers and drivers into the robot arm to reduce communication distance, we applied a Serial Peripheral Interface (SPI) bus between a robot controller and FPGA drivers, designed a communication system, and simplified transmitted data packets that contained synchronous and asynchronous transmissions such as each's position, velocity, and current.

Regarding SPI driver design, Linux provides SPI bus libraries; however, as a non-real-time operating system, Linux resulted in an SPI communication frequency at 1 kHz, a jitter longer than 10,000 μs , and an average latency of 10 μs (Figure 13 [top]). The Linux drivers exhibited excessively long latency and thus were not appropriate for robot systems requiring 1-ms control response. To achieve real-time control, this study used Xenomai's RTDM skin to design a real-time SPI driver, which had a jitter shorter than 100 μs and an average latency of 0.5 μs (Figure 13 [bottom]). These features indicated the SPI driver was appropriate for application in the communication architecture of a real-time controller.

5.3. Real-Time Program Architecture

Focusing on controllers' processing efficiency, we have separated their program tasks according to their level of priority of real-time control. The non-real-time Linux kernel was responsible for low priority tasks such as kinematic calculations, trajectory generation, and user interface display. The Xenomai kernel was responsible for real-time SPI bus communications, control command transmission, state information feedback, and synchronization of each joint's motor state with the drivers with a control period of 1 ms.

A bridge program between the two kernels was designed to facilitate inter-process communication; namely, programs in the two kernels could operate independently during different periods, and their program design could be developed independently. An FPGA-based DSP control system was used to conduct 1-ms servo control and dynamic force compensation. In the first joint, SPI communication was

used between its driver and the robot controller, and RS485 serial communication was used between its driver and other joints' drivers. The overall communication and control architecture is illustrated in Figure 14.

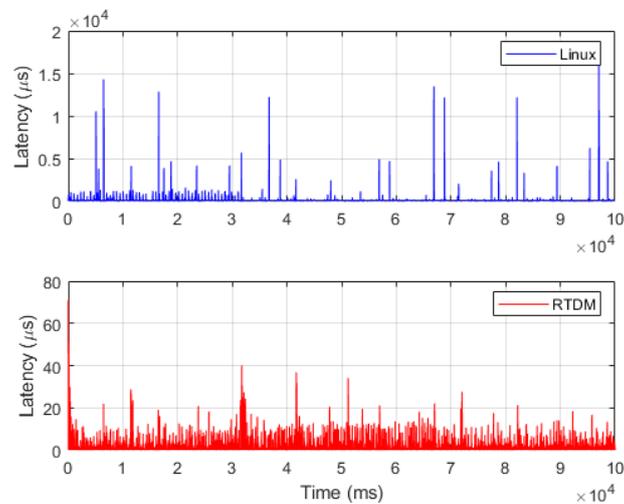


Figure 13. Latency performances of Serial Peripheral Interface (SPI) driver.

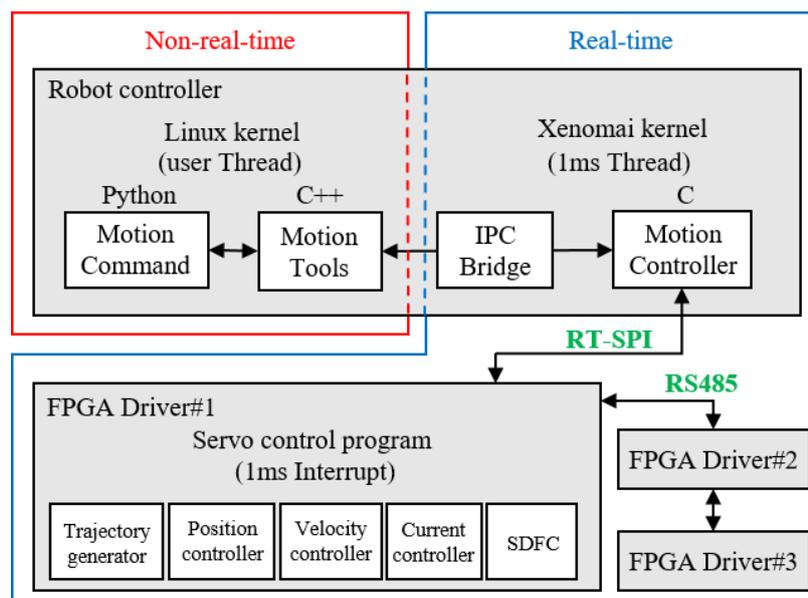


Figure 14. Real-time control architecture.

6. Experiment and Results

This study used a self-developed seven-axis robot arm, which exhibited concise design with low-functionality components, to conduct experiments to verify the validity of this study. The specifications and prices of the main components used in the robot arm's joints are listed in Table 7.

Table 7. Prices and specifications of the main joint components.

Item	Type	Source	Unit Price
Motor	3-phase BLDC motor	China	35 USD
Reduction gear	Harmonic driver	Taiwan	400 USD
Driver	FPGA build-in DSP	Taiwan	100 USD
Total price for 7 axis robot arm			5000 USD

6.1. Joint Position Accuracy Analysis

Different motors and reduction ratios used in the joints resulted in their various minimum resolutions (Table 2), with each of the joints achieving a position resolution of less than 0.2° . To verify the control effectiveness of the SDFC, we conducted experiments to test both single-axis and multiple-axis position accuracy.

6.1.1. Experiment 1: Single-Axis Repeatability Accuracy

Within each joint's working range (Table 8), we conducted position control by randomly generating 20 positions under an acceleration duration of 0.5 s, deceleration duration of 1 s, and speed range of 20–100% (an interval of 10% and maximum angular velocity of $60^\circ/\text{s}$). Figure 15 depicts the position errors of each joint, and Table 8 presents the mean absolute errors (MAE) and standard deviations (SD) of each joint.

Table 8. Results of experiment 1 and experiment 2.

Joint	Work Range		Experiment 1		Experiment 2	
	Min.	Max.	MAE	SD	MAE	SD
J1	-90°	90°	0.11°	0.12°	0.06°	0.06°
J2	-30°	90°	0.23°	0.33°	0.23°	0.24°
J3	-90°	90°	0.09°	0.10°	0.08°	0.07°
J4	-90°	90°	0.11°	0.10°	0.20°	0.07°
J5	-90°	-90°	0.08°	0.09°	0.12°	0.20°
J6	-90°	90°	0.08°	0.09°	0.08°	0.06°
J7	-360°	360°	0.08°	0.10°	0.09°	0.17°

MAE = Mean Absolute Error. SD = Standard Deviation.

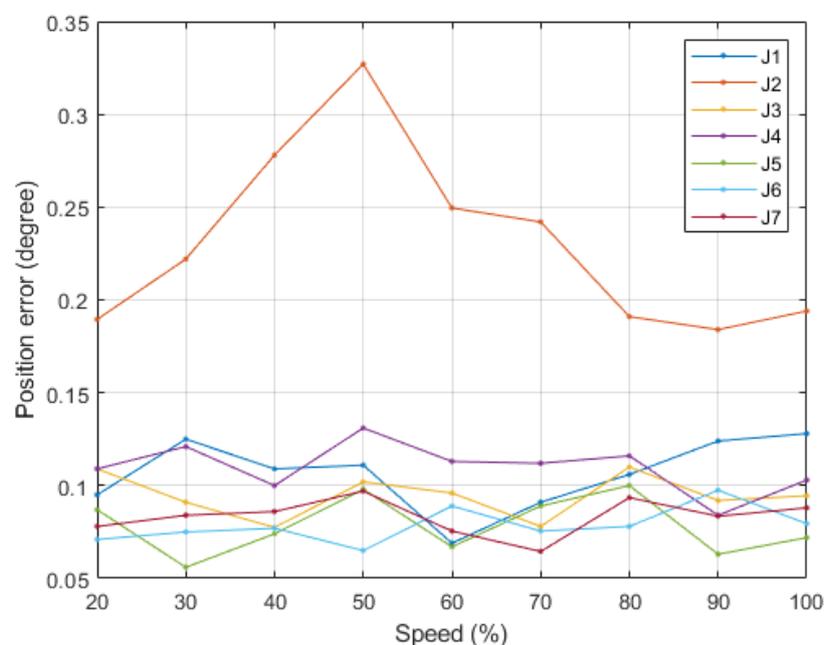


Figure 15. Position error results of experiment 1.

6.1.2. Experiment 2: Multiple-Axis Repeatable Accuracy

The following four positions with different loads were selected:

$$P1 = [0^\circ, -30^\circ, 0^\circ, -60^\circ, 0^\circ, -90^\circ, 0^\circ]$$

$$P2 = [30^\circ, -60^\circ, 30^\circ, -60^\circ, 30^\circ, -60^\circ, 60^\circ]$$

$$P3 = [0^\circ, -90^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ]$$

$$P4 = [60^\circ, -60^\circ, 30^\circ, -90^\circ, -45^\circ, -40^\circ, -60^\circ]$$

Figure 16 shows these four positions' resulting robot arm postures (P1 to P4). This study used linear interpolation to generate each joint's trajectories, according to which we posed the robot arm in the four positions under a maximum velocity of 60°/s and recorded the steady-state position errors. The process of position control and recording of steady-state position errors has been repeated for 100 cycles. Figure 17 demonstrates mean absolute errors of each position; Table 8 shows the mean absolute errors and standard deviations of each joint.

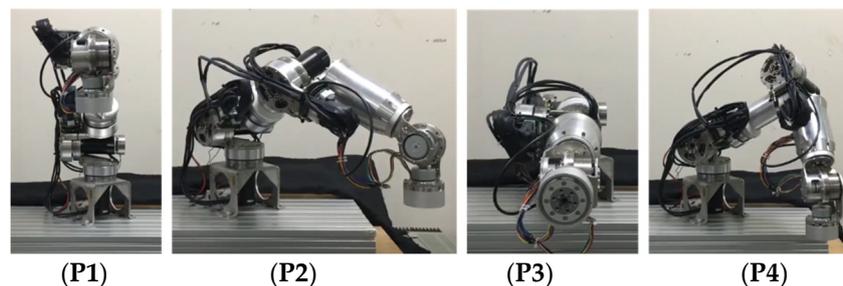


Figure 16. Robot arm postures using multiple-axis control.

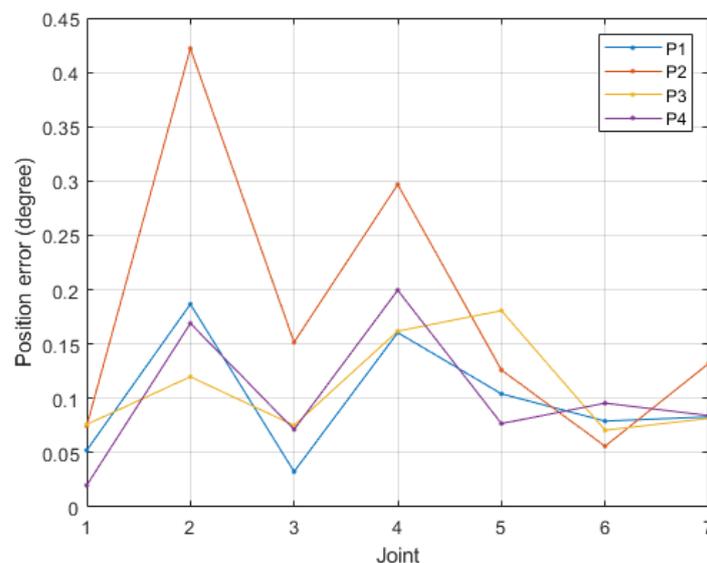


Figure 17. Position error results of experiment 2.

According to Experiment 1, Joint 2 exhibited the lowest accuracy and had an error of 0.23°, which was larger than the minimum resolution (0.1875°). This is because, of all joints, Joint 2 had the highest load, which impeded its fine movements and position convergence, thus failing to effectively eliminate steady-state errors. All joints other than Joint 2 had errors of approximately 0.1°, which was smaller than the minimum resolution under different velocity controls.

In Experiment 2, because the inertia and gravity in each joint changed concurrently, mutual force between them occurred, and thus the position errors were larger than those of single-axis control. In particular, Joints 2 and 4 had the largest steady-state errors; this result is because the two joints carried the largest loads of all. According to the standard deviations in Table 8, all joint position errors were within 0.5° (about 95% of data fall within two standard deviations of the mean), equal to 1~2

times the minimum resolution. The results of two experiments verified that the SDFC helped the low-gain controllers reduce dynamic error under different motor speeds and loads.

6.2. Compensator Efficiency Analysis

According to the study's control structure, the motor actual current output was the sum of servo controllers' and the SDFC's torque, as shown in Figure 6. High similarity between a compensator's calculated dynamic force and the actual force received indicates a reduced servo control output current produced by errors. Therefore, the servo control current can be seen in the SDFC compensating for error; a low proportion of servo current in the total output current (16) indicated high accuracy of the SDFC, where E is current error rate, I_{servo} is the servo controller current, and I_{out} is the total motor output current.

$$E = \left| \frac{I_{\text{servo}}}{I_{\text{out}}} \right| \times 100\% \quad (16)$$

Experiment 3: Compensator Efficiency Analysis

This experiment separated the motion process into acceleration, period I; constant velocity, period II; and deceleration, period III (Figure 18). The acceleration period involved friction, gravity, and inertia torque; the constant velocity involved only friction and gravity torque. The current error rate (E) in the three periods were used to analyze the accuracy of compensators' force calculation relative to actual force received.

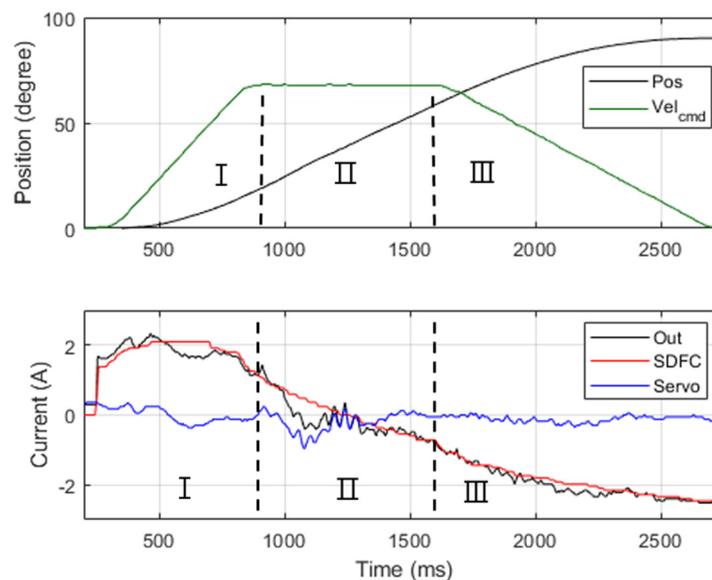


Figure 18. Dynamic current performance for Joint 2.

The test condition for each joint comprised a velocity of $60^\circ/\text{s}$ and a position control range of 0° to 90° . During the experiments, we recorded each joint's total output current, the SDFC's currents, and the servo controllers' currents; subsequently, we calculated the current error rate (E) from (16) and average currents in the three motion periods to evaluate the efficiency of the compensator. Table 9 presents the experimental results.

According to the results of Experiment 3, Joint 2 had a relatively large servo current output in the constant velocity period and exhibited a current error (E) of approximately 40%. This is because the controlled current crossed the zero point during period II (Figure 18), which produced large oscillations in current control. Accordingly, improving current control in the zero-crossing point can possibly reduce such control errors.

Table 9. Results of experiment 3.

Joint	Current Error Rate E [%]			Current [A]	
	I	II	III	Servo	Total
J1	8.56	2.01	2.18	0.07	1.67
J2	10.67	39.68	6.08	0.17	1.51
J3	10.64	8.12	4.48	0.16	2.32
J4	12.12	4.49	4.99	0.13	1.88
J5	7.26	1.51	6.49	0.05	0.91
J6	11.80	4.34	3.71	0.06	1.07
J7	8.22	4.00	3.99	0.08	1.41

Among the three motion periods, Period I demonstrated the highest current error rate (>10%) because the static friction must be overcome to activate the motor, and the friction compensator as (10) cannot accurately calculate the static friction for motor starting; therefore, a large control error occurred. The remaining periods all exhibited a current error rate of less than 10%, indicating that the accuracy of SDFC in force calculation relative to the actual forces received was higher than 90%; therefore, the SDFC had adequate resolution for robot arm dynamic control.

According to the result in Table 9, all joints' average servo currents were smaller than 0.2 A, which is less than 10% of the total current output. The result verified that the force compensation provided by the SDFC in dynamic control could correct more than 90% of all errors, whereas other errors, such as calculation errors and system model errors, could be corrected using closed-loop servo controllers. The integrated use of a dynamic force compensator and servo controllers provided favorable control effectiveness without the use of a high-precision system model.

7. Conclusions

This study used sensorless drivers and SDFCs to design and control a seven-axis robot arm. Through open-loop current compensation, we solved the servo controllers' delay problem caused by an insufficient number of sensors and the problems of insufficient force and slow response caused by low-gain control. The SDFC was designed by simplifying calculations and conducting experiments to obtain the needed parameters without the high-precision system model, which overcame problems occurring when using only Hall-effect sensors, and simplified the dynamic control calculation steps to achieve more programmable calculation with the use of low-cost processors. Moreover, because the compensator is separate from the servo controllers, it can be applied to various servo systems and robot arms with only some modifications in the system model parameters, thus exhibiting high flexibility of use and wide applicability. The compensator could also reduce the system loads of servo control as well as the performance requirements of a motor driver and sensor, thus reducing the development cost of robot arms. The results presented 0.2° accuracy of joint position control and 90% real-time current compensation from SDFCs; these performances were good enough for service application. In addition, motors with many poles, such as Maxon Motor's flat motors with 11 poles, can be used to increase position resolution from the 0.188° achieved by this study to approximately 0.068° . Overall, this study verified the safe and accurate control of the proposed sensorless, low-gain driver and thus its applicability in service robots.

Author Contributions: Conceptualization, C.-Y.L. and Y.-C.L.; methodology, S.-H.Y. and P.-C.T.; software, S.-H.Y. and P.-C.T.; validation, S.-H.Y.; formal analysis, S.-H.Y. and P.-C.T.; investigation, S.-H.Y. and Y.-C.L.; resources, P.-C.T. and Y.-C.L.; data curation, S.-H.Y.; writing—original draft preparation, S.-H.Y.; writing—review and editing, C.-Y.L. and Y.-C.L.; supervision, C.-Y.L. and Y.-C.L.; project administration, Y.-C.L.; funding acquisition, C.-Y.L. and Y.-C.L.

Funding: This work was financially supported by both the Taiwan Building Technology Center and the Center for Cyber-Physical System Innovation from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. Additionally, this work was also financially supported by Ubiquilife Technology Corporation in Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IFR International Federation of Robotics. Executive Summary World Robotics 2018 Service Robots. 2018. Available online: https://ifr.org/downloads/press2018/Executive_Summary_WR_Service_Robots_2018.pdf (accessed on 1 June 2019).
2. Vinod, B.; Wood, L. Service Robots: The Next Big Productivity Platform. PwC Technology Forecast. 2016. Available online: <http://usblogs.pwc.com/emerging-technology/service-robots-the-next-big-productivity-platform> (accessed on 1 June 2019).
3. Albu-Schäffer, A.; Haddadin, S.; Ott, C.; Stwmme, A.; Wimböck, T.; Hirzinger, G. The DLR lightweight robot: Design and control concepts for robots in human environments. *Ind. Robot Int. J.* **2007**, *34*, 376–385. [[CrossRef](#)]
4. Albu-Schäffer, A.; Haddadin, S.; Ott, C. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *Int. J. Robot. Res.* **2007**, *26*, 23–39. [[CrossRef](#)]
5. Knežević, J.M. Low-cost low-resolution sensorless positioning of dc motor drives for vehicle auxiliary applications. *IEEE Trans. Veh. Technol.* **2013**, *62*, 4328–4335. [[CrossRef](#)]
6. Haines, G.; Ertugrul, N. Wide speed range sensorless operation of brushless permanent-magnet motor using flux linkage increment. *IEEE Trans. Ind. Electron.* **2016**, *63*, 4052–4060. [[CrossRef](#)]
7. Chen, S.; Zhou, X.; Bai, G.; Wang, K.; Zhu, L. Adaptive commutation error compensation strategy based on a flux linkage function for sensorless brushless DC motor drives in a wide speed range. *IEEE Trans. Power Electron.* **2017**, *33*, 3752–3764. [[CrossRef](#)]
8. Song, X.; Han, B.; Zheng, S.; Fang, J. High-precision sensorless drive for high-speed BLDC motors based on the virtual third harmonic back-EMF. *IEEE Trans. Power Electron.* **2017**, *33*, 1528–1540. [[CrossRef](#)]
9. Li, T.; Zhou, J. High-Stability Position-Sensorless Control Method for Brushless DC Motors at Low Speed. *IEEE Trans. Power Electron.* **2018**, *34*, 4895–7903. [[CrossRef](#)]
10. Fryman, J.; Matthias, B. Safety of industrial robots: From conventional to collaborative applications. In Proceedings of the ROBOTIK 7th German Conference, Munich, Germany, 21–22 May 2012; pp. 1–5.
11. Plooij, M.; Wolfslag, W.; Wisse, M. The effect of the choice of feedforward controllers on the accuracy of low gain controlled robots. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 4090–4097.
12. Gaz, C.; Flacco, F.; De Luca, A. Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–5 June 2014; pp. 1386–1392.
13. Cano, R.V. Towards a Model-Based Motion Control Design for a 7-axis Robotic Arm LWA4D by Schunk. Master's Thesis, UMEA University, Umea, Sweden, 2016.
14. Yen, S.H.; Tang, P.C.; Lin, Y.C.; Lin, C.Y. Development of a virtual force sensor for a low-cost collaborative robot and applications to safety control. *Sensors* **2019**, *19*, 2603. [[CrossRef](#)] [[PubMed](#)]
15. Cham, C.L.; Samad, Z.B. Brushless dc motor electromagnetic torque estimation with single-phase current sensing. *J. Elect. Eng. Technol.* **2014**, *9*, 866–872. [[CrossRef](#)]
16. Khalil, W.; Dombre, E. *Modeling, Identification and Control of Robots*, 3rd ed.; Taylor and Francis Group: New York, NY, USA, 2002; pp. 191–194.
17. Ridwan, M.; Yuniarto, M.N. Electrical equivalent circuit based modeling and analysis of brushless direct current (BLDC) motor. In Proceedings of the 2016 IEEE International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, Indonesia, 28–30 July 2016; pp. 471–478.
18. Ding, L.; Wu, H.; Yao, Y.; Yang, Y. Dynamic model identification for 6-DOF industrial robots. *J. Robot.* **2015**, *471478*. [[CrossRef](#)]
19. Chitta, S.; Sucan, I.; Cousins, S. Moveit! [ROS topics]. *IEEE Robot. Autom. Mag.* **2012**, *19*, 18–19. [[CrossRef](#)]
20. Spong, M.W.; Hutchinson, S.; Vidyasagar, M. *Robot Modeling and Control*; Wiley: New York, NY, USA, 2006; Volume 3.
21. Wolf, S.; Iskandar, M. Extending a dynamic friction model with nonlinear viscous and thermal dependency for a motor and harmonic drive gear. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 783–790.

22. Verdonck, W.; Swevers, J. Improving the dynamic accuracy of industrial robots by trajectory pre-compensation. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 11–15 May 2002; pp. 3423–3428.
23. Yabuki, A.; Ohishi, K.; Miyazaki, T.; Yokokura, Y. Force control including contact process using acceleration-sensor-based instantaneous state observer for high-stiffness gear drive. In Proceedings of the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), Santa Clara, CA, USA, 8–10 June 2016; pp. 651–656.
24. Huang, C.C.; Lin, C.H.; Wu, C.K. Performance evaluation of Xenomai 3. In Proceedings of the 17th Real-Time Linux Workshop (RTLWS), Graz, Austria, 21–22 October 2015.
25. Hambarde, P.; Varma, R.; Jha, S. The survey of real time operating system: RTOS. In Proceedings of the 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, Nagpur, India, 9–11 January 2014; pp. 34–39.
26. Kiszka, J. The real-time driver model and first applications. In Proceedings of the 7th Real-Time Linux Workshop, Lille, France, 3–4 November 2005.
27. Boterenbrood, H. *CANopen High-Level Protocol for CAN-Bus*; Nikhef: Amsterdam, The Netherlands, 2005.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).