

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

171,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Hierarchical Sliding Mode Control for a 2D Ballbot That Is a Class of Second-Order Underactuated System

*Dinh Ba Pham, Soon-Geul Lee, Thi Hang Bui
and Tien Phat Truong*

Abstract

2D Ballbot is an actual under-actuated system with second-order nonholonomic velocity constraints and input coupling case where only control input is employed to control two outputs of the system. Controlling such a system is not easy because it faces many changelings including nonlinearities, external disturbances, and uncertainties. This study proposed a robust control system for a Ballbot mobile robot. The proposed control scheme is constructed using the hierarchical sliding mode control technique. The kinematics and dynamics of the Ballbot are derived. A Lyapunov function is used to prove the stability of the closed-loop control system. The stabilizing and transferring problems are investigated through several simulations and experiments by using the actual Ballbot platform.

Keywords: sliding mode control, Ballbot, under-actuated system, Lyapunov analysis, stabilizing and transferring

1. Introduction

Ballbot has a body that balances on a single spherical wheel (ball). The robot uses a drive mechanism consisting of three omnidirectional wheels (OWs) for a ball [1] to ensure both stabilizing and transferring. The Ballbot can free travel in any direction on plat plane, even if the robot is crowded with people in a common human-coexisting environment.

An inherent property of the Ballbot system is a naturally nonlinear underactuated multi-input multi-output system, in which the number of the control input signal is less than the number of output signals. This property introduces the challenge related to Ballbot control design. Several control designs were accomplished using a simplified model (linear model) to overcome the complexities of the mathematical equations. The linear model is the first way to deal with the complexity of the mathematical equation of the Ballbot system. Reference [2] designed a control system based on a

state feedback controller to control stabilizing and transferring. The experimental results showed the ability to stabilize and travel with loads in any direction. However, it can be led only to small external disturbances. Studies in [3, 4] employed a double-loop linear control system to stabilize a Ballbot. Their control system had two controllers: a PI inner-loop controller and a linear quadratic regulator (LQR) outer-loop controller. Kantor et al. [5] proposed a double-loop control system. Both inner-loop and outer-loop controllers were PID controllers. Sukvichai and Parnichkun [6] used a linear controller based on an LQR control scheme for a double-level Ballbot.

The second way to deal with the complexity of the mathematical equation of the Ballbot system is to use nonlinear and intelligent controls such as partial feedback linearization control (FLC), sliding mode control (SMC), or fuzzy control. Lotfiani et al. [7] employed an SMC controller along fuzzy trajectory planning to control a Ballbot to track the desired trajectory. Moreover, a collocated PFL method [8] was introduced for a Ballbot. The open-loop trajectory generation with the collocated PFL controller was simulated which showed a low position error.

Some researchers have studied intelligent control techniques. Reference [9] suggested an intelligent tracking control system combined with a dual Mamdani-type fuzzy control strategy and supervisory control technique for an omnidirectional spherical mobile platform. The experiment results showed the position tracking response of the robot. A fuzzy wavelet cerebellar-model-articulation controller [10] was proposed for a team of multiple Ballbots.

SMC is a well-known and robust nonlinear control scheme. To enhance robustness for both actuated [11] and under-actuated [12–19] systems, several controllers have been employed the SMC control method. The application of SMC for Ballbot control can be found in several previous studies. Ching-Wen et al. [20] introduced a hierarchical SMC based on backstepping to control stabilizing and agile trajectory tracking of a Ballbot with exogenous disturbance. Reference [21] enhanced Ching-Wen's backstepping SMC using interval type 2 fuzzy neural networks for motion control of a Ballbot with a four-motor inverse mouse ball-diving mechanism. However, these studies only presented numerical simulations.

As mentioned, the Ballbot is naturally an under-actuated system in which the number of the control inputs is less than the number of the outputs. The Ballbot system is nonlinear. Moreover, the control design problem also faces uncertainties, un-modeling, and disturbances. Thus, the SMC control scheme is a suitable approach for the Ballbot system.

In this study, we designed a robust controller to control a Ballbot. The proposed control scheme is a basis of the hierarchical SMC technique to execute two main tasks including stabilizing the body on the top of the ball and maneuvering the ball on the floor.

2. System model

2.1 Kinematic model

The general kinematics of the Ballbot is a relationship between $\dot{\phi}_k = [\dot{\phi}_x \ \dot{\phi}_y \ \dot{\phi}_z]^T$, the angular velocity of the ball relative to the body frame, and $\dot{\psi}_i$ ($i = 1, 2, 3$), the angular velocities of the three OWs relative to the reference frame of the body. The kinematics relationship of dual-row OWs was discussed in [22]. For the Ballbot, the

three OWs, a kind of single-row OW, are mounted on the body of the robot. The driving axes are separated by 120° in the x - y plane and each driving axis is sloped downward by α .

The main assumptions for simplification of the system are:

- i. Body and floor are absolute solids.
- ii. No slippage occurs on the surfaces of the ball-floor and the ball-OW.
- iii. The points of contact of the ball-ground and the OWs-ball are points of contact.

The non-slip condition of the surface of the ball-OWs is met by requiring that the projection of velocities of the ball at all contact points in the actuation direction of the OW be the same and expressed mathematically as

$$\dot{\psi}_i r_w = (\dot{\phi}_k \times \mathbf{P}_i) \mathbf{u}_{wi}, i = 1, 2, 3, \quad (1)$$

where \mathbf{P}_i is the position vector from the center of the ball to the contact point i between the ball and the i th OW in the body frame as indicated in **Figure 1**, \mathbf{u}_{wi} is the vector of direction cosines of the i th OW contact point velocities in the actuation directions, and r_w is the radius of the OW.

Therefore, the angular velocities of the OWs are expressed as

$$\dot{\psi}_i = \frac{1}{r_w} (\dot{\phi}_k \times \mathbf{P}_i) \mathbf{u}_{wi}, i = 1, 2, 3. \quad (2)$$

As shown in **Figure 1**, the position vector of the contact point \mathbf{P}_i can be defined by

$$\mathbf{P}_1 = [r_k \sin \alpha \quad 0 \quad r_k \cos \alpha]^T, \quad (3a)$$

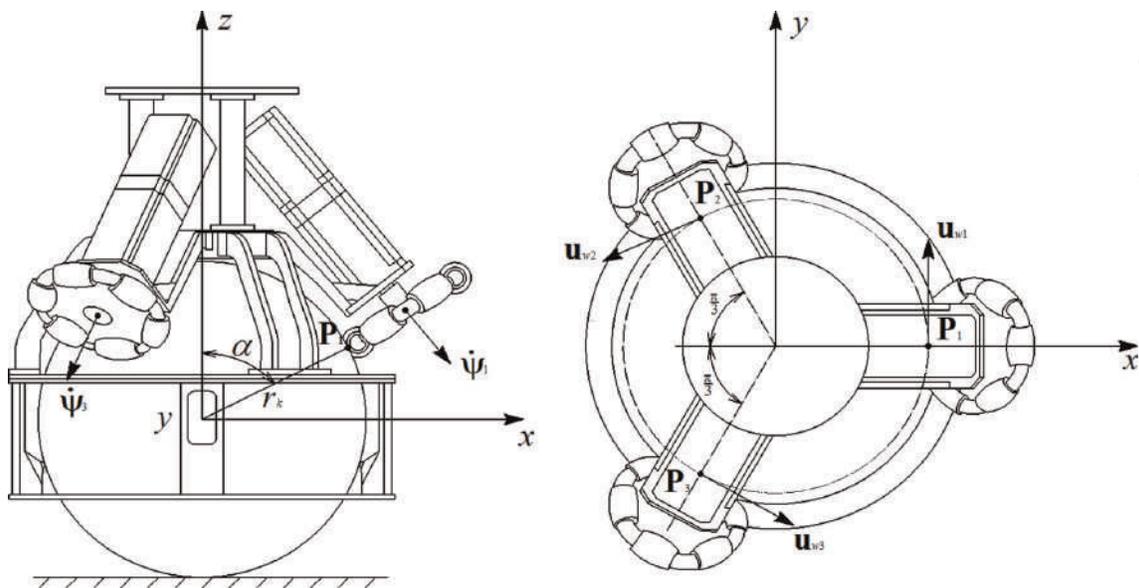


Figure 1.
 Sketch of decomposed angular velocities of three OWs and a ball.

$$\mathbf{P}_{2,3} = \begin{bmatrix} -\frac{1}{2}r_k \sin \alpha & \pm \frac{\sqrt{3}}{2}r_k \sin \alpha & r_k \cos \alpha \end{bmatrix}^T. \quad (3b)$$

Furthermore, the direction cosine vectors that provide the corresponding direction of the speed at the contact point between the OWs and the ball are defined by

$$\mathbf{u}_{w1} = [0 \quad 1 \quad 0]^T, \quad (4a)$$

$$\mathbf{u}_{w2,3} = \begin{bmatrix} \mp \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \end{bmatrix}^T. \quad (4b)$$

Substituting (3a), (3b), (4a) and (4b) into (2) yields

$$[\dot{\varphi}_x \quad \dot{\varphi}_y \quad \dot{\varphi}_z]^T = \mathbf{J}(\alpha)[\dot{\psi}_1 \quad \dot{\psi}_2 \quad \dot{\psi}_3]^T, \quad (5)$$

where $\mathbf{J}(\alpha)$ is the Jacobian matrix of the system

$$\mathbf{J}(\alpha) = \begin{bmatrix} -\frac{2r_w}{3r_k \cos \alpha} & \frac{r_w}{3r_k \cos \alpha} & \frac{r_w}{3r_k \cos \alpha} \\ 0 & -\frac{\sqrt{3}r_w}{3r_k \cos \alpha} & \frac{\sqrt{3}r_w}{3r_k \cos \alpha} \\ \frac{r_w}{3r_k \sin \alpha} & \frac{r_w}{3r_k \sin \alpha} & \frac{r_w}{3r_k \sin \alpha} \end{bmatrix}. \quad (6)$$

The Jacobian matrix (6) represents the effect of the zenith angle on the motion relationship between the ball and three single-row OWs.

2.2 Dynamic model

Figure 2 presents the planar model of the Ballbot. The system of the Ballbot is modeled as a rigid cuboid on top of a ball to get a simple calculation of the system parameters. The ball mass m_k and OW mass m_w are considered as point masses concentrated at their centers. m_a represents the equivalent mass of all components including the body, drive mechanism, and other parts. Chosen generalized coordinates of the Ballbot system include $x_k(t)$, $y_k(t)$, $\theta_x(t)$, and $\theta_y(t)$, here, $x_k(t)$, $y_k(t)$ denote

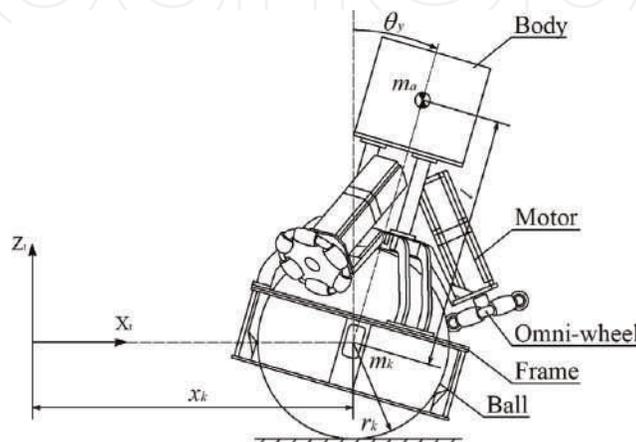


Figure 2.
Planar model of the main driving system of the Ballbot.

the position of the, $\theta_x(t)$, and $\theta_y(t)$ are the orientation of the body around the x - and y -axes. The control input is the resultant moment of the three actuators, whose x and y terms are τ_x and τ_y .

The Ballbot system dynamics are described by obtaining the kinetic and potential of the ball, body, and OWs. For the ball,

$$T_{kx} = \frac{1}{2} \left(m_k + \frac{I_k}{r_k^2} \right) \dot{y}_k^2, V_{kx} = 0, \quad (7)$$

where r_k is the ball radius and I_k is the ball momentum inertia.

The OWs are attached to the body of the Ballbot. Thus, only the rotational motion of the OWs has to be calculated, that is,

$$T_{wx} = \frac{3I_w \cos^2 \alpha}{4r_w^2} (\dot{y}_k + r_k \dot{\theta}_x)^2, V_{wx} = 0, \quad (8)$$

where r_w is the radius of each OW, α represents the zenith angle, and I_w indicates the momentum inertia of each OW.

For the body,

$$T_{ax} = \frac{1}{2} I_x \dot{\theta}_x^2 + \frac{1}{2} m_a (\dot{y}_k - l \dot{\theta}_x \cos \theta_x)^2 + \frac{1}{2} m_a l^2 \dot{\theta}_x^2 \sin^2 \theta_x, V_{ax} = m_a g l \cos \theta_x, \quad (9)$$

where I_x is the body momentum inertia around the x -axis, l represents the distance from the ball center to the body mass center, and g denotes the gravitational acceleration.

In the y - z plane, the generalized coordinates of the Ballbot system are defined as $\mathbf{q}_x = [y_k \ \theta_x]^T$ and the friction vector

$$\mathbf{D}(\dot{\mathbf{q}}_x) = [b_y \dot{y}_k \ b_{rx} \dot{\theta}_x]^T, \quad (10)$$

where b_y and b_{rx} are the viscous damping coefficients that model the ball-floor friction and ball-body friction in the y - z plane, respectively.

The Lagrangian function L_x is expressed as

$$\begin{aligned} L_x &= T_x - V_x = T_{kx} + T_{wx} + T_{ax} - (V_{kx} + V_{wx} + V_{ax}) \\ &= \left(\frac{1}{2} \left(m_k + \frac{I_k}{r_k^2} \right) \dot{y}_k^2 + \frac{3I_w \cos^2 \alpha}{4r_w^2} (\dot{y}_k + r_k \dot{\theta}_x)^2 \right. \\ &\quad \left. + \frac{1}{2} I_x \dot{\theta}_x^2 + \frac{1}{2} m_a (\dot{y}_k - l \dot{\theta}_x \cos \theta_x)^2 + \frac{1}{2} m_a l^2 \dot{\theta}_x^2 \sin^2 \theta_x - m_a g l \cos \theta_x \right), \end{aligned} \quad (11)$$

The Euler-Lagrange equation that describes the motion of the Ballbot is

$$\frac{d}{dt} \left(\frac{\partial L_x}{\partial \dot{\mathbf{q}}_x} \right) - \frac{\partial L_x}{\partial \mathbf{q}_x} = \frac{1}{r_w} \begin{bmatrix} 1 \\ r_k \end{bmatrix} \tau_x - \mathbf{D}(\dot{\mathbf{q}}_x). \quad (12)$$

From (12), the dynamic equations on the y - z plane are expressed as follows:

$$\ddot{y}_k a_1 + (a_4 - a_3 \cos \theta_x) \ddot{\theta}_x + a_3 \dot{\theta}_x^2 \sin \theta_x + b_y \dot{y}_k = r_w^{-1} \tau_x, \quad (13)$$

$$(a_4 - a_3 \cos \theta_x) \ddot{y}_k + \ddot{\theta}_x a_2 + b_{rx} \dot{\theta}_x - a_5 \sin \theta_x = r_k r_w^{-1} \tau_x, \quad (14)$$

where $a_1 = m_k + \frac{I_{kx}}{r_k^2} + m_a + \frac{3I_w \cos^2 \alpha}{2r_w^2}$, $a_2 = m_a l^2 + \frac{3I_w r_k^2 \cos^2 \alpha}{2r_w^2} + I_x$, $a_3 = m_a l$, $a_4 = \frac{3I_w \cos^2 \alpha}{2r_w^2} r_k$, $a_5 = m_a g l$, I_k refers to the ball momentum inertia, I_w is the OW momentum inertia, I_x denotes the body momentum inertia about the x -axis, b_y and b_{rx} are the viscous damping coefficients.

Then, (13) and (14) can be rewritten as follows:

$$\ddot{y}_k = F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x1}(\mathbf{q}_x) \tau_x, \quad (15)$$

$$\ddot{\theta}_x = F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x2}(\mathbf{q}_x) \tau_x, \quad (16)$$

where $F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x) = A_x^{-1} \left[(a_3 \cos \theta_x - a_4) (a_5 \sin \theta_x - b_{rx} \dot{\theta}_x) - a_2 (a_3 \dot{\theta}_x^2 \sin \theta_x + b_y \dot{y}_k) \right]$,

$$G_{x1}(\mathbf{q}_x) = A_x^{-1} r_w^{-1} (a_2 + a_3 r_k \cos \theta_x - a_4 r_k),$$

$F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x) = A_x^{-1} \left[(a_4 - a_3 \cos \theta_x) (a_3 \dot{\theta}_x^2 \sin \theta_x + b_y \dot{y}_k) + a_1 (a_5 \sin \theta_x - b_{rx} \dot{\theta}_x) \right]$,

$$G_{x2}(\mathbf{q}_x) = A_x^{-1} r_w^{-1} (a_3 \cos \theta_x - a_4 + a_1 r_k),$$

$$A_x = a_1 a_2 - (a_4 - a_3 \cos \theta_x)^2.$$

In the x - z plane, the mathematical equations describe the Ballbot system dynamics as follows:

$$\ddot{x}_x b_1 + (b_4 \cos \theta_y - b_3) \ddot{\theta}_y - b_4 \dot{\theta}_y^2 \sin \theta_y + b_{xx} \dot{x}_k = -r_w^{-1} \tau_y, \quad (17)$$

$$(b_4 \cos \theta_y - b_3) \ddot{x}_k + \ddot{\theta}_y b_2 - b_5 \sin \theta_y + b_{ry} \dot{\theta}_y = r_k r_w^{-1} \tau_y, \quad (18)$$

where $b_1 = m_k + \frac{I_k}{r_k^2} + \frac{3I_w \cos^2 \alpha}{2r_w^2} + m_a$, $b_2 = \frac{3I_w r_k^2 \cos^2 \alpha}{2r_w^2} + m_a l^2 + I_y$, $b_3 = \frac{3I_w \cos^2 \alpha}{2r_w^2} r_k$, $b_4 = m_a l$, $b_5 = m_a g l$, I_y represents the body momentum inertia about the y -axis, b_x and b_{ry} are the viscous damping coefficients.

In the x - z plane, minimal coordinates of the Ballbot system denotes $\mathbf{q}_y = [x_k \ \theta_y]^T$, then (17) and (18), are rewritten as follows:

$$\ddot{x}_k = F_{y1}(\mathbf{q}_y, \dot{\mathbf{q}}_y) + G_{y1}(\mathbf{q}_y) \tau_y, \quad (19)$$

$$\ddot{\theta}_y = F_{y2}(\mathbf{q}_y, \dot{\mathbf{q}}_y) + G_{y2}(\mathbf{q}_y) \tau_y, \quad (20)$$

where $F_{y1}(\mathbf{q}_y, \dot{\mathbf{q}}_y) = A_y^{-1} \left[b_2 (b_4 \sin \theta_y \dot{\theta}_y^2 - b_{xx} \dot{x}_k) + (b_3 - b_4 \cos \theta_y) (b_5 \sin \theta_y - b_{ry} \dot{\theta}_y) \right]$,

$$G_{y1}(\mathbf{q}_y) = -A_y^{-1} r_w^{-1} (b_2 - b_3 r_k + b_4 r_k \cos \theta_y),$$

$F_{y2}(\mathbf{q}_y, \dot{\mathbf{q}}_y) = A_y^{-1} \left[(b_3 - b_4 \cos \theta_y) (b_4 \sin \theta_y \dot{\theta}_y^2 - b_{xx} \dot{x}_k) + b_1 (b_5 \sin \theta_y - b_{ry} \dot{\theta}_y) \right]$,

$$G_{y2}(\mathbf{q}_y) = A_y^{-1} r_w^{-1} (r_k b_1 - b_3 + b_4 \cos \theta_y),$$

$$A_y = b_1 a_2 - (b_4 \cos \theta_y - b_3)^2.$$

3. Control system design

3.1 Hierarchical sliding mode control design

It can be assumed that all state variables are measurable. To design this controller in the y - z plane, first we introduce a suitable pair of SMSs

$$\begin{cases} s_{x1} = c_{x1} e_{x1} + \dot{e}_{x1} \\ s_{x2} = c_{x2} e_{x2} + \dot{e}_{x2} \end{cases}, \quad (21)$$

where c_{x1} and c_{x2} are positive constants, e_{x1} and e_{x2} are tracking errors

$$\begin{cases} e_{x1} = y_k - y_{kd} \\ e_{x2} = \theta_x - \theta_{xd} \end{cases}, \quad (22)$$

where $y_{kd} = \text{const}$ denotes the desired position of the ball and θ_{xd} is the desired tilt angle about the x -axis of the body.

When the Ballbot balances, it means that the desired tilt angle $\theta_{xd} = 0$. Then, (21) can be rewritten as

$$\begin{cases} s_{x1} = c_{x1} (y_k - y_{kd}) + \dot{y}_k \\ s_{x2} = c_{x2} \theta_x + \dot{\theta}_x \end{cases}. \quad (23)$$

Let $\dot{s}_{x1} = 0$ and $\dot{s}_{x2} = 0$, the equivalent control laws of the two subsystems can be gotten as

$$\tau_{xeq1} = -G_{x1}^{-1}(\mathbf{q}_x) [c_{x1} \dot{y}_k + F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x)], \quad (24)$$

$$\tau_{xeq2} = -G_{x2}^{-1}(\mathbf{q}_x) [c_{x2} \dot{\theta}_x + F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x)]. \quad (25)$$

The hierarchical SMC law is deduced as follows. The first layer SMS is defined as $S_{x1} = s_{x1}$. For the first layer SMS, the SMC law and the Lyapunov function are defined as

$$\tau_{x1} = \tau_{xeq1} + \tau_{xsw1}, \quad (26)$$

and

$$V_{x1}(t) = 0.5 S_{x1}^2, \quad (27)$$

where τ_{xsw1} is the switch control part of the first layer SMC. Differentiate $V_{x1}(t)$ with respect to time t

$$\dot{V}_{x1}(t) = S_{x1} \dot{S}_{x1}. \quad (28)$$

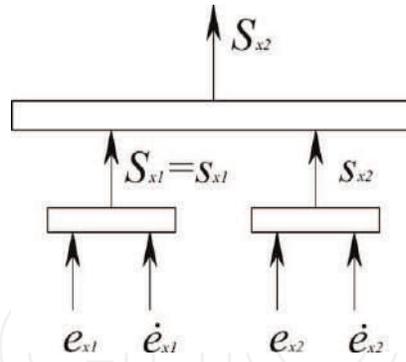


Figure 3.
Structure of hierarchical sliding mode surfaces.

Let

$$\dot{S}_{x1} = -k_{x1}S_{x1} - \eta_{x1}\text{sign}(S_{x1}), \quad (29)$$

where k_{x1} and η_{x1} are positive constants.

The first layer SMC law can be deduced from Eqs. (26) and (27), that is,

$$\tau_{x1} = \tau_{xeq1} + G_{x1}^{-1}(\mathbf{q}_x)\dot{S}_{x1}. \quad (30)$$

The second layer SMS is constructed based on the first layer SMS S_1 and s_2 , as shown in **Figure 3**.

$$S_{x2} = \alpha_x S_{x1} + s_{x2}, \quad (31)$$

where α_x is the sliding mode parameter.

For the second layer SMS, the SMC law and the Lyapunov function are defined as

$$\tau_{x2} = \tau_{x1} + \tau_{xeq2} + \tau_{xsw2}, \quad (32)$$

and

$$V_{x2}(t) = 0.5S_{x2}^2, \quad (33)$$

where τ_{xsw2} is the switch control part of the second layer SMC.

Differentiating $V_{x2}(t)$ with respect to time t yields

$$\dot{V}_{x2}(t) = S_{x2}\dot{S}_{x2}. \quad (34)$$

Let

$$\dot{S}_{x2} = -k_{x2}S_{x2} - \eta_{x2}\text{sign}(S_{x2}), \quad (35)$$

where k_{x2} and η_{x2} are positive constants.

The total control law of the presented hierarchical SMC can be deduced as follows:

$$\tau_{x2} = \frac{\alpha_x G_{x1}(\mathbf{q}_x)\tau_{xeq1} + G_{x2}(\mathbf{q}_x)\tau_{xeq2} + \dot{S}_{x2}}{\alpha_x G_{x1}(\mathbf{q}_x) + G_{x2}(\mathbf{q}_x)}. \quad (36)$$

Similarly, the total control law of the hierarchical SMC in the x - z plane also given as

$$\tau_{y2} = \frac{\alpha_y G_{y1}(\mathbf{q}_y) \tau_{yeq1} + G_{y2}(\mathbf{q}_y) \tau_{yeq2} + \dot{S}_{y2}}{\alpha_y G_{y1}(\mathbf{q}_y) + G_{y2}(\mathbf{q}_y)}. \quad (37)$$

3.2 Stability analysis

Theorem 1: If considering the total control law (36) and the SMSs (23) and (31) for the system dynamics (15) and (16), then SMSs, S_{x1} and S_{x2} are asymptotically stable.

Proof: Integrating both sides (34) with respect to time obtains

$$\int_0^t \dot{V}_{x2} d\tau = \int_0^t (-\eta_{x2} |S_{x2}| - k_{x2} S_{x2}^2) d\tau, \quad (38)$$

Then

$$V_{x2}(t) - V_{x2}(0) = \int_0^t (-\eta_{x2} |S_{x2}| - k_{x2} S_{x2}^2) d\tau, \quad (39)$$

It can be found that

$$V_{x2}(0) = V_{x2}(t) + \int_0^t (\eta_{x2} |S_{x2}| + k_{x2} S_{x2}^2) d\tau \geq \int_0^t (\eta_{x2} |S_{x2}| + k_{x2} S_{x2}^2) d\tau. \quad (40)$$

Therefore, it can be achieved that

$$\lim_{t \rightarrow \infty} \int_0^t (\eta_{x2} |S_{x2}| + k_{x2} S_{x2}^2) d\tau \leq V_{x2}(0) < \infty.$$

By using Barbalat's lemma [23], we can obtain that if $t \rightarrow \infty$ then $\eta_{x2} |S_{x2}| + k_{x2} S_{x2}^2 \rightarrow 0$. Then, $\lim_{t \rightarrow \infty} S_{x2} = 0$.

By applying Barbalat's lemma, we can get $\lim_{t \rightarrow \infty} S_{x2} = 0$.

Thus, both S_{x1} and S_{x2} are asymptotically stable.

Theorem 2: If considering the control law (36) and the SMSs of (23) for the system dynamics (15) and (16), then SMSs, s_{x1} and s_{x2} , are also asymptotically stable.

Proof: From Theorem 1, the SMS of the ball subsystem dynamics is asymptotically stable.

Now, we will prove that the SMS of the body subsystem dynamics is asymptotically stable. Limiting of both sides of (31) obtains

$$\lim_{t \rightarrow \infty} S_{x2} = \lim_{t \rightarrow \infty} (\alpha_x S_{x1} + s_{x2}) = \alpha_x \left(\lim_{t \rightarrow \infty} S_{x1} \right) + \lim_{t \rightarrow \infty} s_{x2} = \lim_{t \rightarrow \infty} s_{x2}. \quad (41)$$

The result of (41) shows $\lim_{t \rightarrow \infty} s_{x2} = \lim_{t \rightarrow \infty} S_{x2} = 0$. It demonstrates that the SMS of the body subsystem dynamics is asymptotically stable. Thus, the all SMSs of the subsystems are asymptotically stable.

Theorem 3: If considering total control law (36) and the SMSs (23) for the system dynamics (15) and (16), assuming that all the state variables are equivalent infinitesimal, then the parameter boundary of the SMS is $0 < c_{x1} < \left| \lim_{\mathbf{x} \rightarrow \mathbf{0}} \left(\frac{F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x)}{\dot{y}_k} \right) \right|$ and $0 < c_{x2} < \left| \lim_{\mathbf{x} \rightarrow \mathbf{0}} \left(\frac{F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x)}{\dot{\theta}_x} \right) \right|$, where $\mathbf{x} = [\mathbf{q}_x \quad \dot{\mathbf{q}}_x]$.

Proof: By solving $s_{xi} = 0$, the lower boundary of c_{xi} can be obtained

$$\begin{cases} s_{x1} = c_{x1}(\dot{y}_k - \dot{y}_{kd}) + \dot{y}_k = 0 \\ s_{x2} = c_{x2}\dot{\theta}_x + \dot{\theta}_x = 0 \end{cases} \quad (42)$$

The eigenvalue of (42) should be negative for meeting the system stability. Thus, the lower boundary of c_{xi} is $c_{xi} > 0$.

The upper boundary of c_{xi} can be gotten from $\dot{s}_{xi} = 0$

$$\begin{cases} \dot{s}_{x1} = c_{x1}\dot{\dot{y}}_k + F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x1}(\mathbf{q}_x)\tau_{xeq1} = 0 \\ \dot{s}_{x2} = c_{x2}\dot{\dot{\theta}}_x + F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x2}(\mathbf{q}_x)\tau_{xeq2} = 0 \end{cases} \quad (43)$$

Therefore,

$$\begin{cases} c_{x1} = \left| \frac{(F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x1}(\mathbf{q}_x)\tau_{xeq1})}{\dot{\dot{y}}_k} \right| \\ c_{x2} = \left| \frac{(F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x) + G_{x2}(\mathbf{q}_x)\tau_{xeq2})}{\dot{\dot{\theta}}_x} \right| \end{cases} \quad (44)$$

Further, it can be got

$$\begin{cases} c_{x1} < \frac{(|F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x)| + |G_{x1}(\mathbf{q}_x)\tau_{xeq1}|)}{|\dot{\dot{y}}_k|} \\ c_{x2} < \frac{(|F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x)| + |G_{x2}(\mathbf{q}_x)\tau_{xeq2}|)}{|\dot{\dot{\theta}}_x|} \end{cases} \quad (45)$$

When the state of the subsystem keeps sliding on its SMS and converges to the neighborhood of the control objective, the system can be treated as an autonomous one. Thus, $\tau_{xeqi} = 0$ and the following inequation is gotten $0 < c_{x1} < \left| \lim_{\mathbf{x} \rightarrow \mathbf{0}} \left(\frac{F_{x1}(\mathbf{q}_x, \dot{\mathbf{q}}_x)}{\dot{y}_k} \right) \right|$

and $0 < c_{x2} < \left| \lim_{\mathbf{x} \rightarrow \mathbf{0}} \left(\frac{F_{x2}(\mathbf{q}_x, \dot{\mathbf{q}}_x)}{\dot{\theta}_x} \right) \right|$.

The block diagram of the control system is shown in **Figure 4**. With the real motor torques τ_1 , τ_2 , and τ_3 as functions of the equivalent torques about the x - and y -axes, τ_x and τ_y , in matrix form, yield [24].

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} \frac{2}{3 \cos \alpha} & -\frac{1}{3 \cos \alpha} & -\frac{1}{3 \cos \alpha} \\ 0 & \frac{\sqrt{3}}{3 \cos \alpha} & -\frac{\sqrt{3}}{3 \cos \alpha} \end{bmatrix}^T \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} \quad (46)$$

The corresponding simulation and experiment results will be given in the subsequent sections.

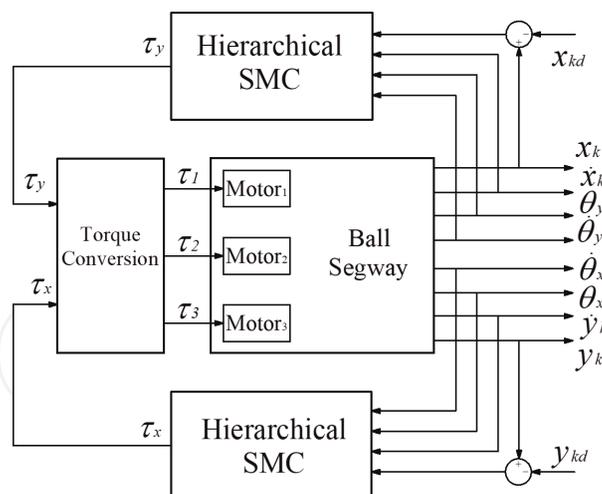


Figure 4.
 Block diagram of the control system.

4. Simulation result

A numerical simulation investigates the stabilize, robustness, and transfer of the proposed controller for the Ballbot. The decoupled dynamics (15) and (16), and (19) and (20) with the proposed control schemes (36) and (37) are modeled in Matlab/Simulink real-time environment with ODE45 and a sampling time of 0.01 seconds.

The parameters of the Ballbot for both simulation and experiment are shown in **Table 1**. For the numerical simulation, the control parameters are tuned by the trial-and-error method and then selected as in **Table 1**.

Various simulations are conducted by considering the stabilizing controls with an initial nonzero tilt angle and with external disturbances and tracking control.

4.1 Stabilizing control with nonzero initial tilt angles

In this simulation, the tilt angles about x - and y -axes are initialized as 6.3° and -6.5° , respectively for checking the behavior. Simulation results are shown in **Figure 5**. The tilt angles responses and the angular velocities are depicted in **Figure 5(a)** and **(b)**, respectively. The position of the ball is shown in **Figure 5(c)**. **Figure 5(d)** shows the curves of the control inputs.

These numerical results demonstrate that the proposed robust controller enables to maintain stabilizing of the Ballbot.

System parameters	Control gains
$m_a = 116\text{kg}$, $I_x = 16.25\text{kgm}^2$, $I_y = 15.85\text{kgm}^2$, $r_w = 0.1\text{m}$, $l = 0.23\text{m}$, $I_w = 0.26\text{kgm}^2$, $r_k = 0.19\text{m}$, $m_k = 11.4\text{kg}$, $I_k = 0.165\text{kgm}^2$, $b_x = b_y = 5\text{Ns/m}$, $b_{rx} = b_{ry} = 3.68\text{Nms/rad}$, $\alpha = 56^\circ$.	$c_{x1} = 0.01$, $c_{x2} = 35$, $\alpha_x = 0.05$, $\eta_{x2} = 0.1$, $k_{x2} = 10$; $c_{y1} = 0.01$, $c_{y2} = 17$, $\alpha_y = 0.05$, $\eta_{y2} = 0.1$, $k_{y2} = 10$

Table 1.
 System parameters and control gains.

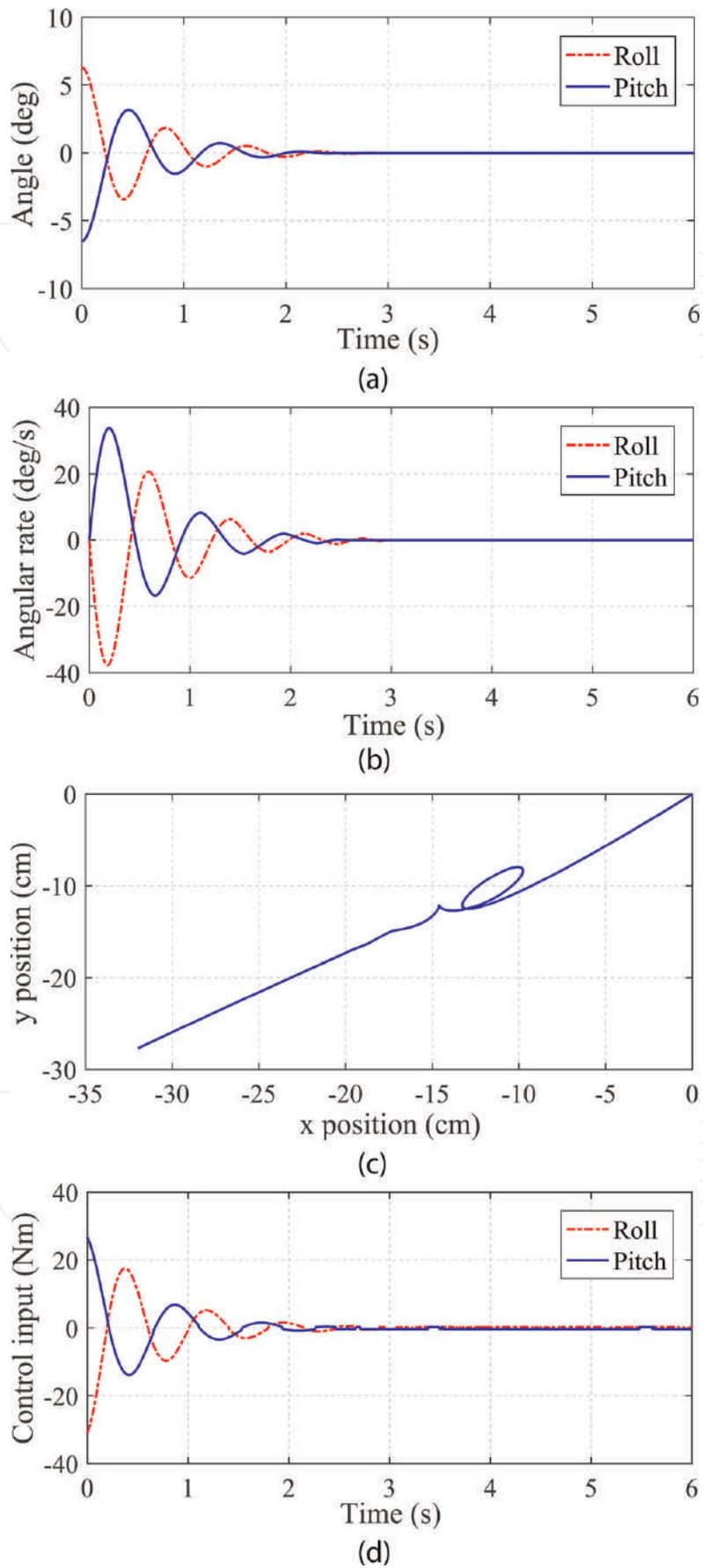


Figure 5. Simulation results of stabilizing while station-keeping. (a) Tilt angles of the body. (b) Angular velocities of the body. (c) Trajectory of the contact point between the ball and the floor. (d) Control inputs.

4.2 Stabilizing control with external disturbances

In the second simulation, the stabilizing control of the Ballbot is investigated under an external disturbance. An external force of 300 N is applied to the Ballbot system at the sixth second while the Ballbot is stabilizing on the floor. As the results, the body tilt angles and angular tilt angles are depicted in **Figure 6(a)** and **(b)**, respectively. Under the external disturbance, The Ballbot cannot maintain its original position of $(x_k, y_k) = (0, 0)$. Instead, the robot has traveled to a new position of $(x_k, y_k) = (-7.4\text{cm}, 3.1\text{cm})$ as shown in **Figure 6(c)**. The control inputs by the proposed control are converged to zeros after 2 seconds, as depicted in **Figure 6(d)**.

4.3 Tracking control

In the third simulation, the Ballbot is commanded to track a rectangular trajectory with a dimension of 75 cm \times 90 cm in 40 s. The system responses of the tilt angles of the body are shown in **Figure 7(b)**. As indicated in **Figure 7(a)**, the proposed control system performs well in tracking desired rectangular trajectory.

5. Experiment result

In this section, several experiments on the actual Ballbot platform (**Figure 8**) are implemented to further verify the performance of the proposed control system. Especially, experiment for the robustness of the controller is executed under external disturbances.

The robot control algorithm is programmed with multithread tasks so that the control period is set to 15 ms. The program also consists of the torque conversion and kinematic model [1] to estimate the position and velocity of the ball.

An inertial measurement unit (IMU) is utilized to measure the orientation and angular rates of the Ballbot. The IMU includes an accelerometer and a gyro sensor. Three encoders with a resolution of 4000 counts/rev are also utilized to obtain the position of the ball. Full state variables of the Ballbot system can be obtained based on the kinematics and the sensor fusion.

The drive mechanism is equipped with brushless DC actuators with a continuous torque of 0.28 Nm and gearboxes with a ratio of 1:4 for driving the ball. Two 48 V lithium battery packs supply power for the actuators and other devices with a working time of several hours.

5.1 Stabilizing control with an initial nonzero tilt angle

This experiment investigates the stabilizing performance of the proposed hierarchical SMC with an actual Ballbot on the flat floor.

The initial position of the Ballbot is set as the origin point and the robot is set at 6.3° in roll angle and - 6.5° in pitch angle. The stabilizing responses of the control system are shown in **Figure 9**. The tilt angles of the body are presented in **Figure 9(a)** in which the steady-state is 1.5 seconds and the steady-state errors of the roll and pitch are 0.4° and 0.5°, respectively. While the angular rates of the body are shown in **Figure 9(b)**. The proposed control system successfully controls the movement of the ball from an origin

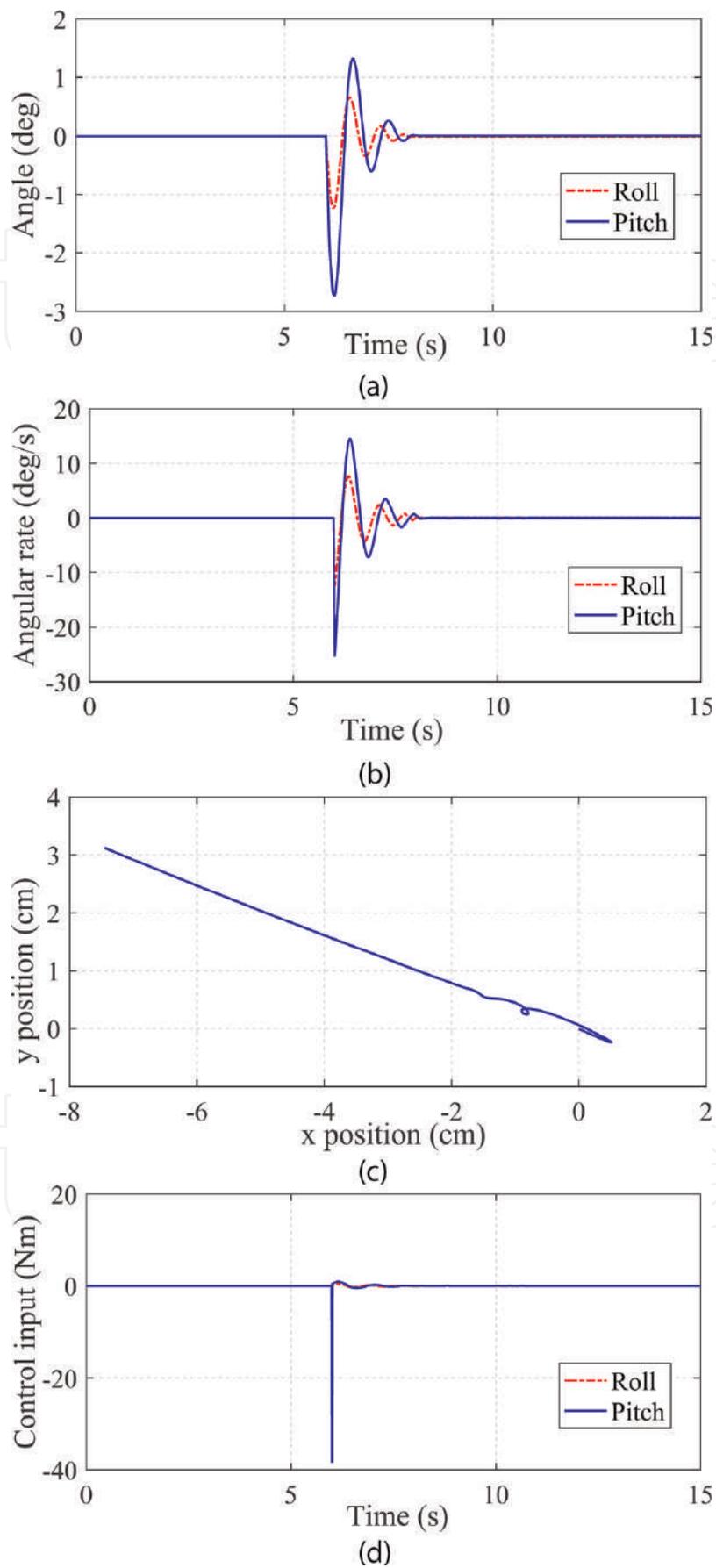


Figure 6. Simulation results of robustness control. (a) Tilt angles of the body. (b) Angular velocities of the body. (c) Trajectory of the contact point between the ball and floor. (d) Control inputs.

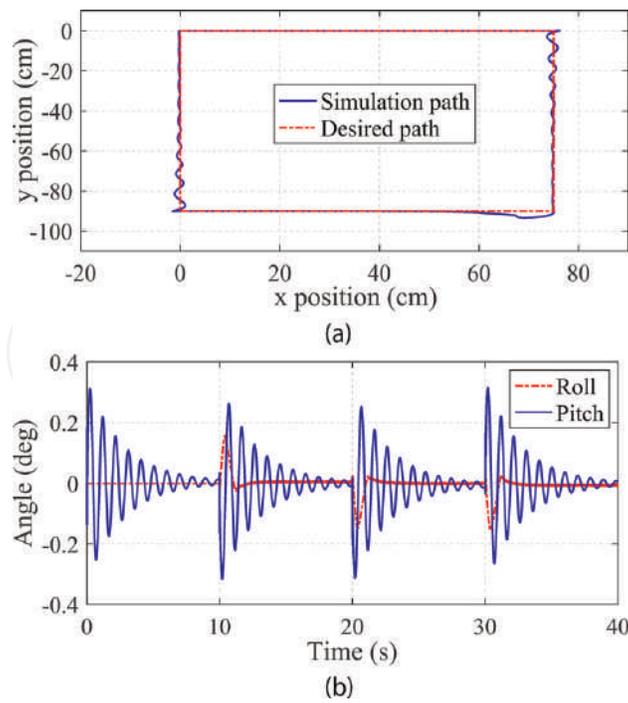


Figure 7. Simulation results of tracking a rectangular path. (a) Trajectory of the contact point between the ball and floor. (b) Tilt angles of the body.

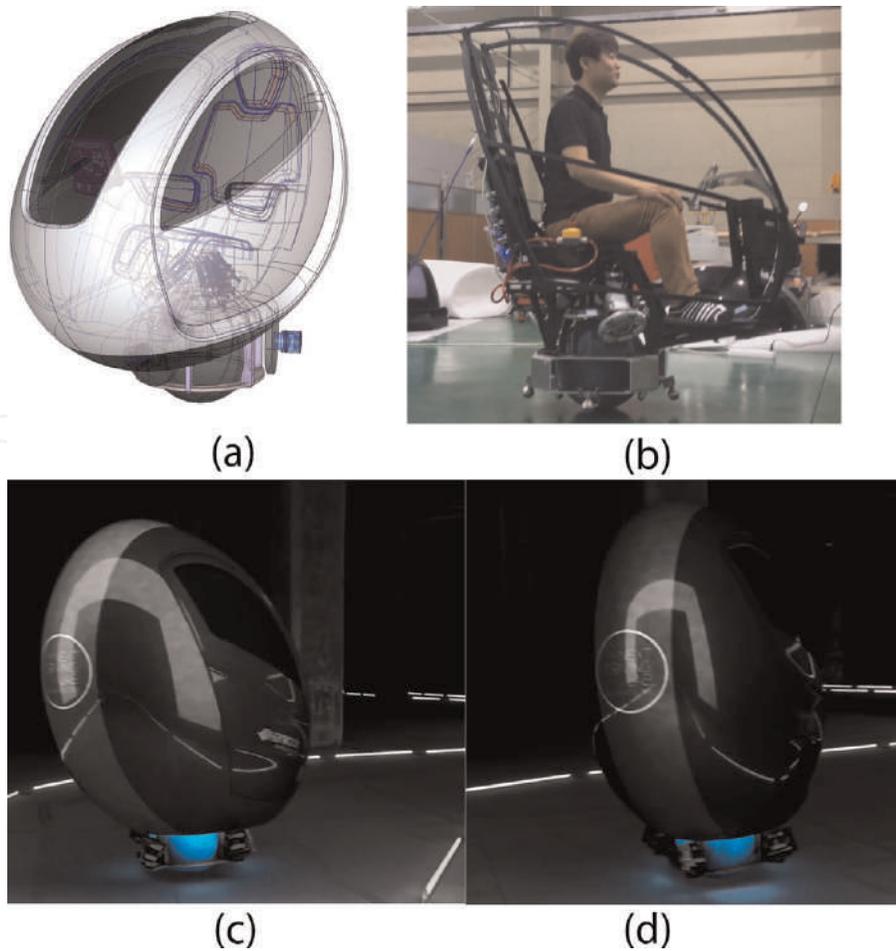


Figure 8. The real Ballbot that is running. (a) Schematic design. (b) Without the cover. (c) Ballbot running. (d) Ballbot running.

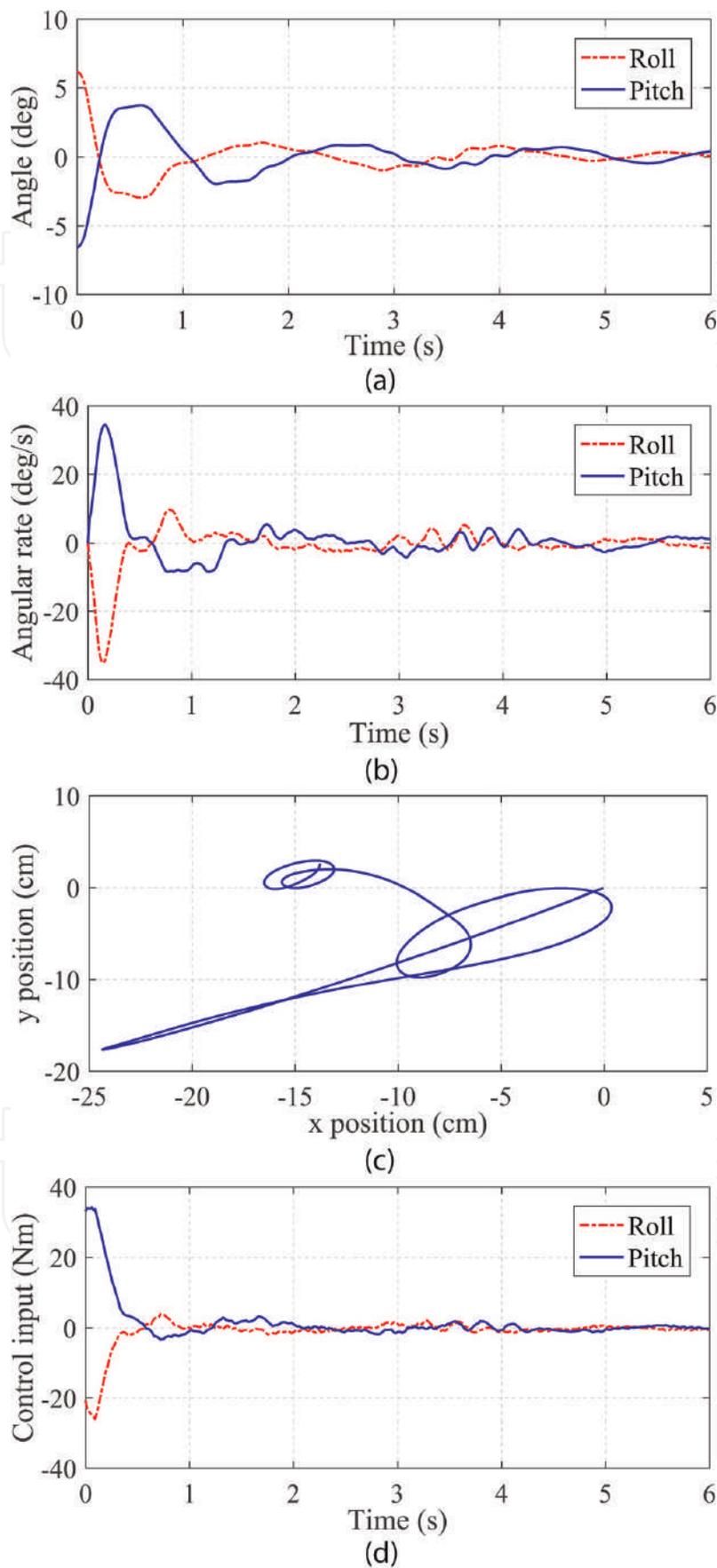


Figure 9. Experiment results of stabilizing and station-keeping. (a) Tilt angles of the body. (b) Angular velocities of the body. (c) Trajectory of the contact point between the ball and floor. (d) Control inputs.

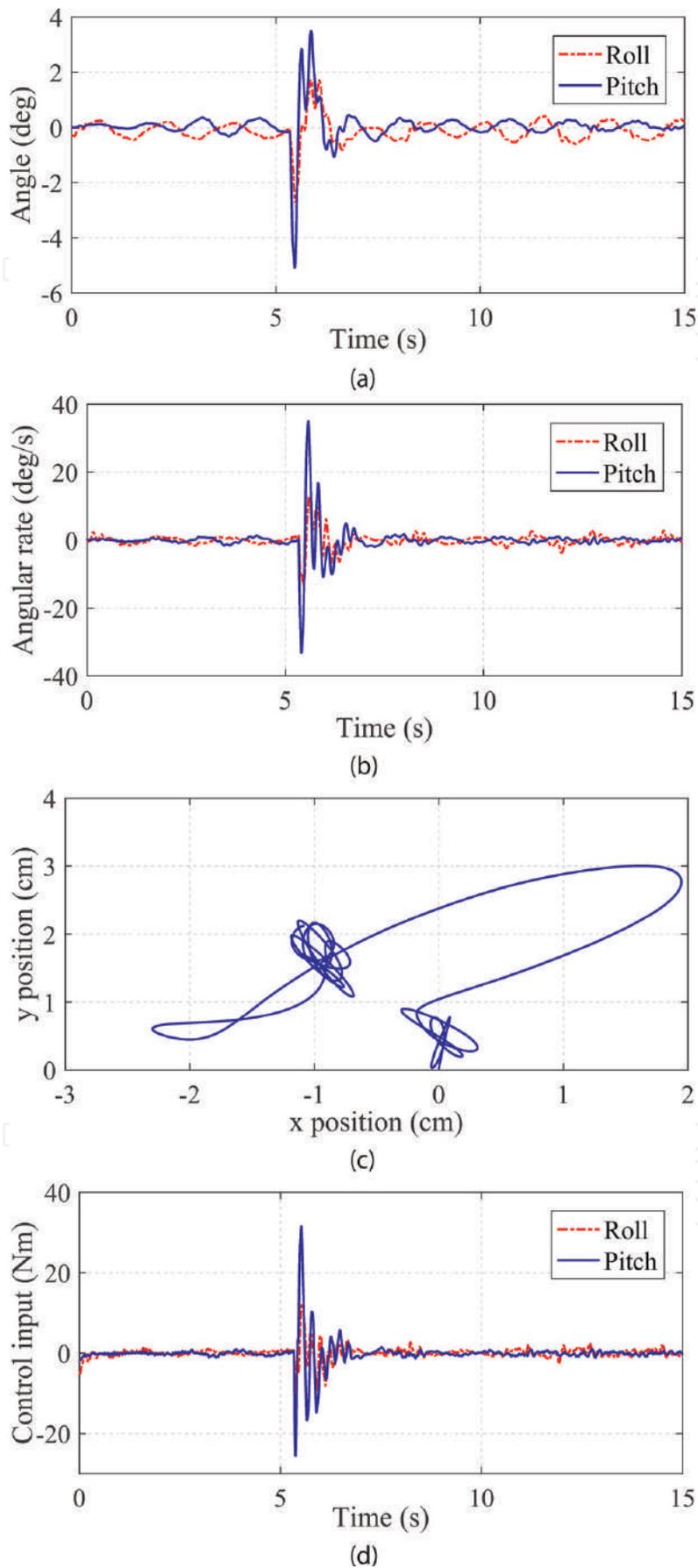


Figure 10. Experiment results of robustness control performance. (a) Tilt angles of the body. (b) Angular velocities of the body. (c) Trajectory of the contact point between the ball and floor. (d) Control inputs.

point to the new point of $(x_k, y_k) = (-14 \text{ cm}, 2 \text{ cm})$ as shown in **Figure 9(c)**. The control input of the proposed control scheme is shown in **Figure 9(d)**.

5.2 Stabilizing control under an external disturbance

The robustness performance of the proposed hierarchical SMC is evaluated by applying the external disturbances to the robot. The experimental scenario is set as: at the beginning, the robot is stabilizing at the origin position. Then the Ballbot is kicked. The amount of the kick is about 300 N.

The tilt angles and angular rate of the body are shown in **Figure 10(a)** and **(b)**. **Figure 10(c)** shows the ball response along the x - and y -axes. Under the kick, the robot moves from the origin position of $(x_k, y_k) = (0, 0)$ to the new position of $(x_k, y_k) = (-0.92 \text{ cm}, 1.65 \text{ cm})$ and then stabilizes at the new position. Torque control input responses are also shown in **Figure 10(d)** to keep the stabilizing of the Ballbot.

5.3 Tracking control

In this experiment, the Ballbot is commanded to track the desired rectangular trajectory with a dimension of $75 \text{ cm} \times 90 \text{ cm}$ within 40 seconds. The system response is presented in **Figure 11**. **Figure 11(a)** shows the trajectory of the ball on the floor.

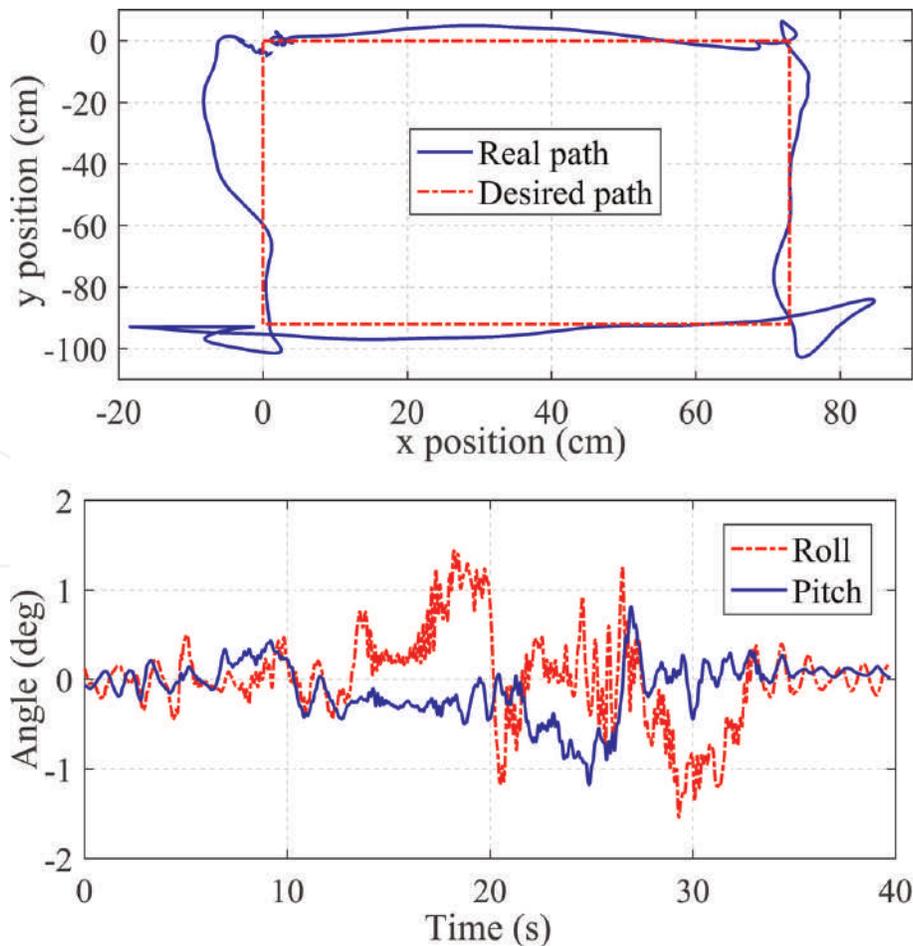


Figure 11. Experimental results of tracking a rectangular path. (a) Trajectory of the contact point between the ball and floor. (b) Tilt angles of the body.

There is some error while the robot tries to track the rectangular desired trajectory. The position error occurs due to uncertainties, an un-modeling system.

The results demonstrate hierarchical SMC behaviors in stabilizing and transferring control of the Ballbot.

6. Conclusions

In this study, a 2D model of the Ballbot is obtained using the Euler-Lagrange equation and the decoupling method. A robust nonlinear controller based on the hierarchical SMC technique is designed for the Ballbot to control stabilizing and transferring. The capability of the closed-loop system with the hierarchical SMC is achieved using the Lyapunov function. The performance and robustness of the hierarchical SMC are examined under several tests in both simulation and experiment. The simulation and experimental results demonstrate the capabilities of the proposed controller for stabilizing and trajectory tracking.

Author details

Dinh Ba Pham^{1*}, Soon-Geul Lee², Thi Hang Bui³ and Tien Phat Truong³

1 Mechanical Engineering Department, Vietnam Maritime University, Haiphong, Vietnam

2 Mechanical Engineering Department, Kyung Hee University, Suwon, South Korea

3 Faculty of Marine Engineering, Vietnam Maritime University, Haiphong, Vietnam

*Address all correspondence to: bapd.vck@vimaru.edu.vn

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Ba PD, Lee SG, Back S, Kim J, Lee MK. Balancing and translation control of a ball Segway that a human can ride. In: 2016 16th International Conference on Control, Automation and Systems (ICCAS). 2016. pp. 477-480
- [2] Kumaga M, Ochiai T. Development of a robot balanced on a ball—Application of passive motion to transport. In: 2009 IEEE International Conference on Robotics and Automation (ICRA'09). 2009. pp. 4106-4111
- [3] Lauwers TB, Kantor GA, Hollis RL. A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006 (ICRA 2006). 2006. pp. 2884-2889
- [4] Ching-Chih T, Ming-Han J, Cheng-Kai C, Ching-Wen L, Siang-Jyun C. Self-balancing and position control using multi-loop approach for ball robots. In: 2010 International Conference on System Science and Engineering (ICSSE). 2010. pp. 251-256
- [5] Kantor G, Hollis H, Nagarajan U. The Ballbot: An omnidirectional balancing mobile robot. *The International Journal of Robotics Research*. 2014;**33**(6):917-930
- [6] Sukvichai K, Parnichkun M. Double-level ball-riding robot balancing: From system design, modeling, controller synthesis, to performance evaluation. *Mechatronics*. 2014;**24**(5):519-532
- [7] Lotfiani A, Keshmiri M, Danesh M. Dynamic analysis and control synthesis of a spherical wheeled robot (Ballbot). In: 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM). 2013. pp. 481-486
- [8] Zabihi H, Talebi HA, Suratgar AA. Open-loop trajectory planning and nonlinear control for underactuated spherical wheel mobile robot (Ballbot). In: 2016 24th Iranian Conference on Electrical Engineering (ICEE). 2016. pp. 549-554
- [9] Chih-Hui C, Wen-Ru T. Design and implementation of an omnidirectional spherical mobile platform. *IEEE Transactions on Industrial Electronics*. 2015;**62**(3):1619-1628
- [10] Ching-Chih T, Chang Hsuan C, Feng-Chun T, Kao-Shing H. Adaptive RFWCMAC cooperative formation control for multiple Ballbots incorporated with coupling dynamics. In: 2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS). 2015. pp. 59-65
- [11] Denker A, Ohnishi K. Robust tracking control of mechatronic arms. *IEEE/ASME Transactions on Mechatronics*. 1996;**1**(2):181-188
- [12] Wang W, Yi J, Zhao D, Liu D. Design of a stable sliding-mode controller for a class of second-order underactuated systems. *IEE Proceedings - Control Theory and Applications*. 2004;**151**(6):683-690
- [13] Yan M-X, Jing Y-W, He Y-G, Sun P. Adaptive sliding mode controller for a class of second-order underactuated systems. In: 2009 Chinese Control and Decision Conference (CCDC'09). 2009. pp. 2782-2786
- [14] Tuan LA, Lee S-GL, Moo S-C. Partial feedback linearization and sliding mode techniques for 2D crane control. *Transactions of the Institute of Measurement and Control*. 2014;**36**(1): 78-87

- [15] Tuan LA, Lee S-G, Nho LC, Kim DH. Model reference adaptive sliding mode control for three dimensional overhead cranes. *International Journal of Precision Engineering and Manufacturing*. 2013; **14**(8):1329-1338
- [16] Tuan LA, Kim J-J, Lee S-G, Lim T-G, Nho LC. Second-order sliding mode control of a 3D overhead crane with uncertain system parameters. *International Journal of Precision Engineering and Manufacturing*. 2014; **15**(5):811-819
- [17] Tuan LA, Lee S-G. Sliding mode controls of double-pendulum crane systems. *Journal of Mechanical Science and Technology*. 2013;**27**(6):1863-1873
- [18] Karkoub MA, Zribi M. Robust control schemes for an overhead crane. *Journal of Vibration and Control*. 2001; **7**(3):395-416
- [19] Almutairi NB, Zribi M. Sliding mode control of a three-dimensional overhead crane. *Journal of Vibration and Control*. 2009;**15**(11):1679-1730
- [20] Ching-Wen L, Ching-Chih T, Yi Yu L, Cheng-Kai C. Dynamic modeling and sliding-mode control of a Ball robot with inverse mouse-ball drive. In: *SICE Annual Conference 2008*. 2008. pp. 2951-2955
- [21] Cheng-Kai C, Ching-Chih T. Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball robot with a four-motor inverse-mouse ball drive. In: *2012 Proceedings of SICE Annual Conference (SICE)*. 2012. pp. 1281-1286
- [22] Weiss A, Langlois RG, Hayes MJD. The effects of dual row omnidirectional wheels on the kinematics of the Atlas spherical motion platform. *Mechanism and Machine Theory*. 2009;**44**(2):349-358
- [23] Slotine JJE, Li W. *Applied Nonlinear Control*. The United States of America: Prentice-Hall, Inc.; 1991. p. 461
- [24] van der Blonk K. Modeling and control of a ball-balancing robot [internship & master thesis]. *Electrical Engineering, Mathematics and Computer Science*, University of Twente, Alten; 2014

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

171,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

PID Gain Tuning for Robust Control of PMDC Motor for External Disturbance Rejection with Constrained Motor Parameter Variations through H_∞

Prasanth Venktareddy, Prashanth Narayanappa Anand and Prakasha Pundareekane Kanchappa

Abstract

This chapter describes the controller modeling for PID gain tuning against the external disturbances with constrained internal parameter variation of the PMDC motor based on an optimization technique of H-infinity. To fit the goals in the H infinite framework, auto tuning of the PID controller gains is used. Different performance goals for tracking are preset as design objectives. Researchers in literature have presented many Robust Control techniques for motor control applications. Methods like back-stepping algorithms, fuzzy and neural based control systems, model predictive control and SMC (Sliding Mode Control) are available in literature. In this chapter, SC (speed control) of PMDC-motor is addressed with variations in outer load disturbances and internal variations of the system parameters for a particular application. C-PID (conventional PID controllers) is preferred, and equivalent robustness characteristics are established using the H-infinity development procedures. The optimization effort is to get simultaneous fast-tracking response and better disturbance rejection.

Keywords: H-infinity, C-PID, PMDC motor, robust control, disturbance rejection

1. Introduction

This necessity of controller design modeling for proportional integral and derivative (PID) gain tuning against the external disturbances with constrained internal parameter variation of the permanent magnet direct current (PMDC) motor based on an optimization technique of H-infinity is highly recommended. Here, PID controller with auto tuning of gains are used to match the goals in H infinity framework. Different performance goals for tracking are preset as design objectives. The speed control of PMDC-motor has so far attracted the attention of the researchers in recent times and many approaches and improvements have been proposed for PMDC motor

drives. PMDC motors are used in electrical equipment, computer peripherals, and manipulators because of their precise speed control capabilities. C-PID controllers have been in use since several years for different applications such as motor-control. The classical tuning methods of PID controller, as well as response method of Zeigler-Nichols frequency considers the system in the mode of oscillation to analyze the tuning procedure [1]. Since the manual tuning of PID controller is not user friendly, as it tends to be a tedious process though being simple in structure.

We know that PID controller fails to address instant tracking / regulation to get robustness against disturbance rejection. The majority of the time, industrial controllers are not properly tuned. Traditionally, motor control applications have better performed in control execution for specific working conditions. Controller parameters can be tuned for exact working conditions with an underlying assumption that the conditions are ordered and defined. Basically, working conditions according to the framework that are prone to variations leads to undesirable results if the variations are not accurately introduced. As these controllers do not guarantee the robustness to inner and outer disturbances, the outer disturbances and the system parameter variations have huge impact on performance and degradation in the applications of motor control.

Researchers in literature have presented many Robust Control techniques for motor control applications. Methods like back-stepping algorithms, fuzzy and neural based control systems, model predictive control and sliding mode control (SMC) are available in literature. The SMC based methodology stimulates the chattering phenomenon due to switching function of the inherent discontinuity. In Eker [2], the authors have demonstrated SMC for various applications of PMDC motor control. In Mamani et al. [3] and Corradini et al. [4], semi-SMC, adaptive SMC and boundary layer control (BLC) have been introduced as the development to classical style of Controlling the SMC and reducing chattering impact. Until now, most of the heuristic engineering methods have been developed to achieve optimal tuning of PID.

PID control with GA optimization based for the DC motor is implemented in Pal et al. [5] and can be stimulated by the aid of normal development methods. These methods have proven that the goal functions have degraded [6].

The PSO (Parasitic Swarm Optimization) is considered as the method of ideal structure to control of BLDC motor by PID control in Nasri et al. [7]. The major advantages being ease of implementation and computational complexity, which facilitates meeting constraints for some of the parameters apart from exhibiting fast convergence.

Position and speed control applications have been implemented using H-infinity-based control [8] providing better robustness in performance against disturbances, making this as an attractive alternative. To develop the Robust Controller, the NN based H-infinity controller was presented in Premkumar et al. [9] that introduces the DC motor H-infinity controller and has addressed parameter uncertainties.

In this chapter, SC application of PMDC-motor is addressed with variations in outer load disturbances and internal variations of the system parameters. Controller is also opted as C-PID and equivalent robustness characteristics are established using the H-infinity development procedures. The optimization effort is to get simultaneous fast-tracking response with better disturbance rejection.

2. PMDC motor model

Using supply current, motor speed and supply voltage relationship, we have the PMDC motor mathematical model as:

Parameter	Nominal value	Variation (tolerable limit)
L	0.00063 H	41%
R	2.07 Ohm	40%
Viscous-friction (K_f)	0.000049 Nms rad ⁻¹	51%
EMF constant (K_b)	0.053 Vs rad ⁻¹	[0.013 to 0.1]
Armature-constant (K_m)	0.053 NmA ⁻¹	[0.012 to 0.1]

Table 1.
 The parameters of motor and their acceptable limits of variation.

$$\frac{di}{dt} = -\frac{R}{L}i - \frac{k_b}{L}\omega + \frac{1}{L}u \quad (1)$$

$$\frac{d\omega}{dt} = \frac{1}{J}k_m i - \frac{1}{J}k_f \omega \quad (2)$$

Here, R and L are resistance and inductance of motor coil, k_b is denoted as the constant of back Electromotive force, J is rotor inertia and k_f is friction constant.

The representation of SS (state space) matrix of similar set of the equations is given by:

$$\frac{d\omega}{dt} \begin{bmatrix} i \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{J} & -\frac{K_f}{J} \end{bmatrix} \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad (3)$$

$$y(t) = [0 \ 1] \begin{bmatrix} i \\ \omega \end{bmatrix} + [0]u(t) \quad (4)$$

The overall transfer function for the speed and input voltage, is expressed as:

$$\frac{\omega(s)}{u(s)} = \frac{K_m}{JLs^2 + (JR + LK_f)s + K_mK_b + RK_f} \quad (5)$$

The transfer function of PMDC motor model is considered for a specified output speed and input voltage provided. The unknown motor parameters are prone to the variations due to aging of motor and its wear and tear. The controller must be built to offer resilience for each motor parameter within acceptable variation limits. The parameters of nominal motor are chosen from the standard motor of Maxon-RE35. The variation limits and motors parameters are given in **Table 1**. The proposed controller design for disturbance rejection in this work is based on the presumption that the PMDC motor internal parameter variations are within tolerable limits as specified for Maxon-RE35. The proposed Robust Controller design is discussed in subsequent section.

3. H ∞ based robust controller design for disturbance rejection

The proposed controller design model should meet the desired specifications within bounds of internal parameter variations. Figure below shows the variation in

the gain of PMDC plant against variations of internal parameters. Similar variations are represented in the domain of frequency via bode plot, that are shown in **Figures 1** and **2**. Henceforth, PID controller is designed to address variation in the model and provide desired specifications for closed loop performance.

This chapter discusses the target specifications for better disturbance rejection and simultaneously set the point tracking. Here, we assume that the control bandwidth, exhibits increasing slope at the crossover frequency compared to disturbance rejection, allowing better gain within the bandwidth. The higher slope is described as lesser phase margin resulting in acceptable overshoot in response to set point. In order to match competing requirements of the tracking rejection and disturbance, the PID controller with 2-DOF (Degree of Freedom) is used as transfer function.

$$u = K_P(br - y) + \frac{K_i}{s}(r - y) + \frac{K_{Ds}}{1 + T_{fs}}(cr - y) \quad (6)$$

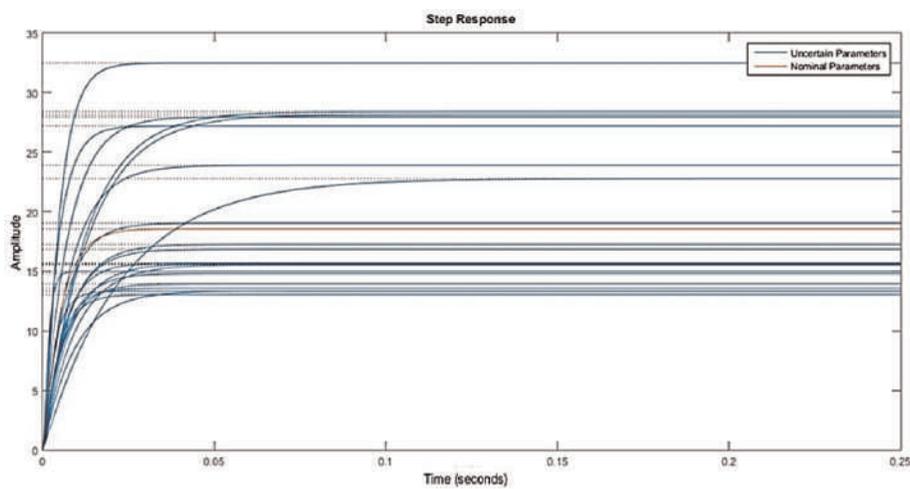


Figure 1.
The variations in PMDC gain of motor plant for uncertain and nominal internal parameter variations.

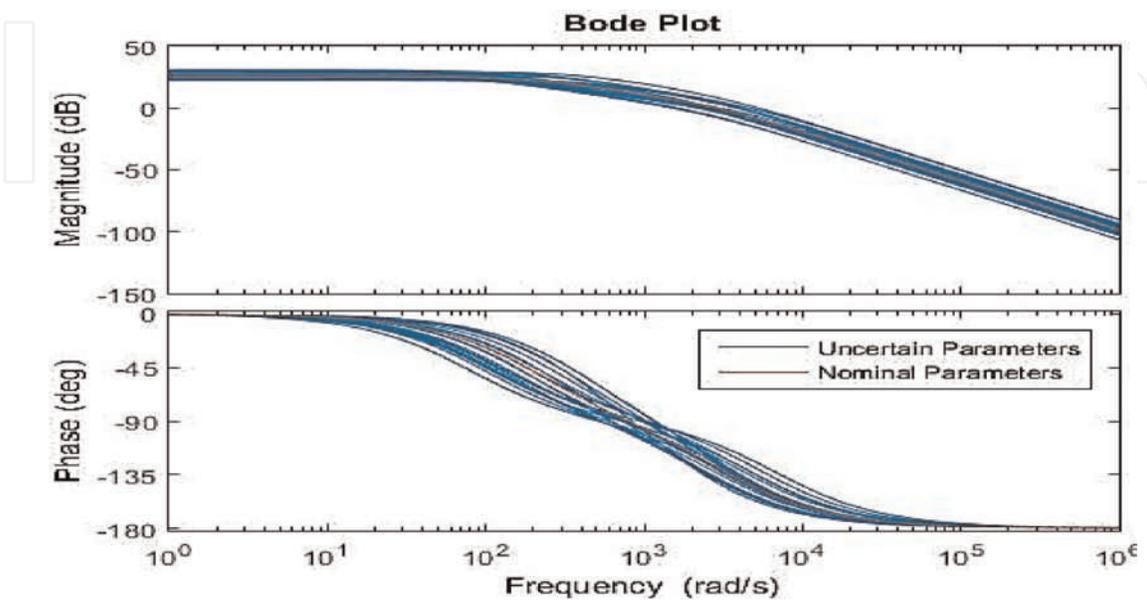


Figure 2.
Bode plot of plant model subjected to the internal parameter variations.

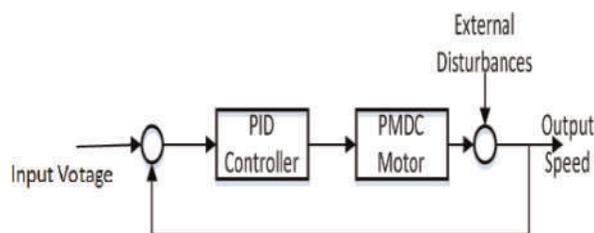


Figure 3.
Block diagram with external disturbances (load variations).

2-DOF (Degree of Freedom) PID controllers contain weighing coefficients associated with derivative and proportional terms. These weighing coefficients facilitate effective disturbance rejection restraining maximization of the over-shoot under optimal conditions. The PID controller of 2-DOF works better in moderating the changes arising in the reference signal or control-signal.

The close loop control system is considered as shown in **Figure 3**. Here, the external disturbances such as load variations on motor shafts and its torque variations are added that influences on the motor parametric variations.

4. The performance goals for the optimization of H_{∞}

The 2-DOF for controller can be tuned by utilizing the approach of H infinity optimization. The open loop gain being a critical indicator of feedback loop behavior, gain of open loop should be more than one in control bandwidth to confirm better DR (Disturbance Rejection) and it must be lower than one outer of control bandwidth to confirm insensitivity to measurement noise of unmodelled dynamics. The ideal performance terms can be displayed regarding execution objectives. To accomplish a decent disturbance rejection and tracking, three execution objectives/limitations are forced on the tuning of controller gain (see discussion in previous section), which is as follows:

- i. "Tracking" – used to identify the RT (response time) to step input
- ii. "ML (Minimum loop) gain" – used to recognize loop gain before the frequency of crossover.
- iii. "ML (Maximum loop) gain" – used to to identify the control bandwidth at higher frequencies.

The CGs (Controller-gains) must be tuned with constraint i.e., function cost, connected with every specification subjected to minimization in H-infinity framework.

4.1 Tracking as an execution objective

The frequency domain specification for monitoring between output and input is described in this performance target. This frequency domain constraint indicates the most extreme relative error as a FF (frequency function). The ME (maximum error) is given by:

$$error_{max} = \frac{(error_{peak})s + \omega_c(error_{dc})}{s + \omega_c} \quad (7)$$

where, ω_c denotes cut-off frequency.

The scalar function $f(x)$ describes the tracking goal, where x is denoted as a tuneable vector of entire parameters in the system. The target optimization (TO) is to modify the parameter function $f(x)$, which is optimized. The scalar function for tracking case is described using $f(x)$:

$$f(x) = \left\| \frac{1}{error_{max}} (T(s, x) - I) \right\|_{\infty} \quad (8)$$

where, $T(s, x)$ is symbolized as closed loop transfer function between input and output.

4.2 Min-LG (minimum) as execution limitation

The minimal gain on the open loop frequency response at specified frequencies is limited by this performance goal. The frequency dependent minimum gain constraint in turn gives the inverse sensitivity function of minimum gain limit. The min constraint gain characterizes the capacity of scalar function $f(x)$ whereas the advancement procedure attempts to drive lower value of $f(x)$. The capacity of scalar $f(x)$ is defined as:

$$f(x) = \|W_S(D^{-1}SD)\|_{\infty} \quad (9)$$

where, W_S is denoted as Min-LG profile and S is defined as sensitivity function.

4.3 Max loop gain as execution limitation

This execution goal aims at highest gain of open loop at the determined frequencies in the given framework. The Max-LG can be characterized as the frequency domain element. This type of constraint restrains upper limit on the corresponding sensitivity. The maximum loop gain determines the scalar function $f(x)$ given by:

$$f(x) = \|W_T(D^{-1}TD)\|_{\infty} \quad (10)$$

where, W_T is symbolized as the reciprocal profile of the Max-LG. T is symbolized as the function of complementary sensitivity.

5. Performance goal description

The three performance goals for the H-infinity minimization such as tracking, maximum LG and minimum LG. The limits/range for these goals is expected as represented in **Figures 4–6** as:

- Tracking better than 2 sec

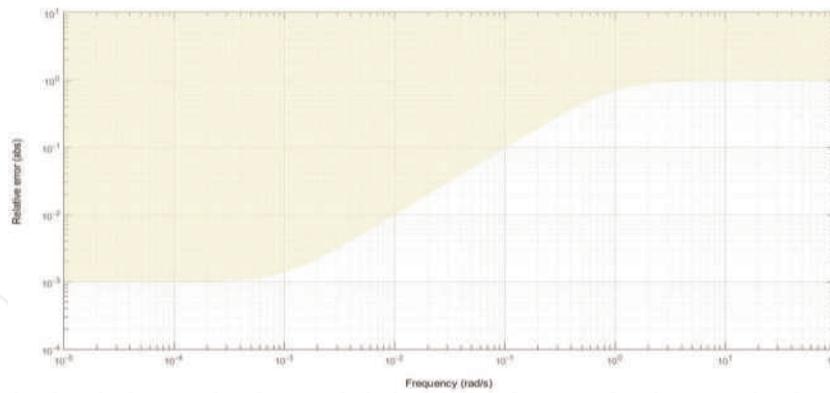


Figure 4.
Performance goal 1: The response for desired tracking.

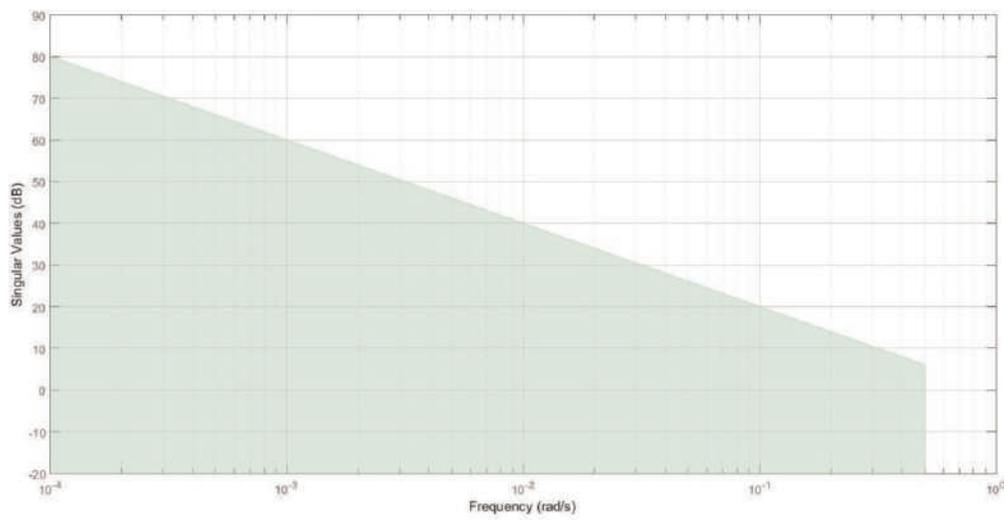


Figure 5.
Performance goal 2: The desired min-LG.

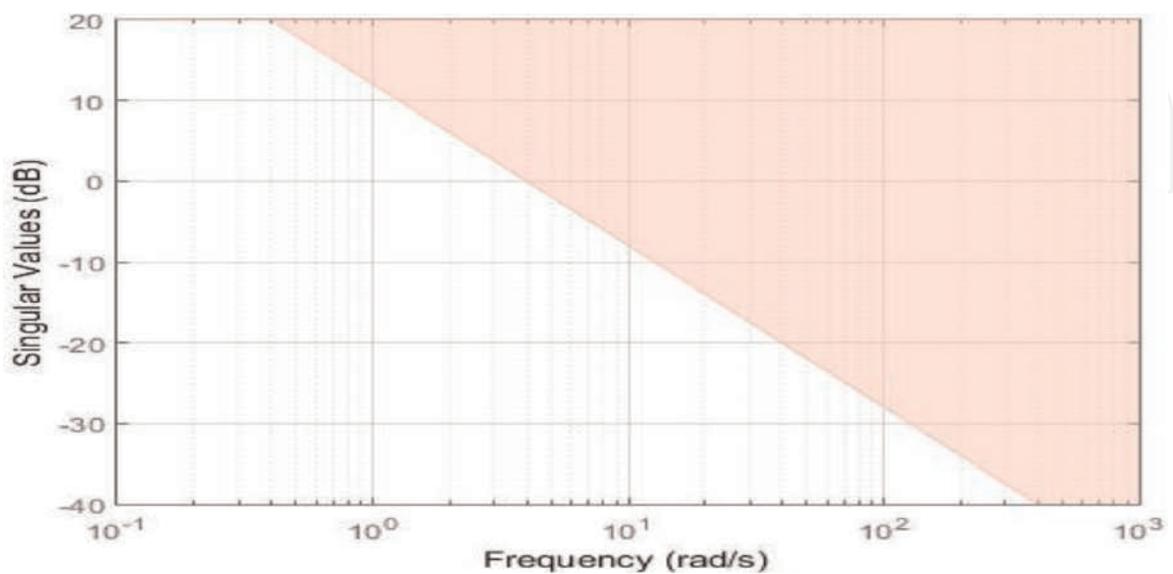


Figure 6.
Performance goal 3: The desired max-LG.

- Min-LG: to be higher below 0.5 rad/s.
- Max-LG: to be less beyond 4 rad/s and the roll off with at least 20db/decade.

6. Simulations and results

The motor model transfer function as described in Eq. (5) along with the constraints from Eqs. (6)–(10) is simulated using MATLAB Simulink.

To model the uncertainty of plant parameters, the specified motor parameters can be used as variables. The PID controller of 2-DOF is characterized to have tunable gains. The analysis point is where disturbance torques are calculated and disturbance sensitiveness is determined. The performance goals as examined in Section 5 are selected and H_∞ minimization of the scalar function relating to the described performance goals is evolved in this work. This results in optimal tuning of controller gains and is tabulated as shown in **Table 2**.

The outcomes are presented in the following figures.

The **Figure 7** represents response of TE (tracking error) of the tuned function of closed loop transfer function. This plot represents the tracking error that is

Parameter	Value
K_p	0.034
T_f	28.447
K_d	-0.942
K_i	0.341

Table 2.

The resulting tuned parameters of 2-DOF PID from simulation for optimization.

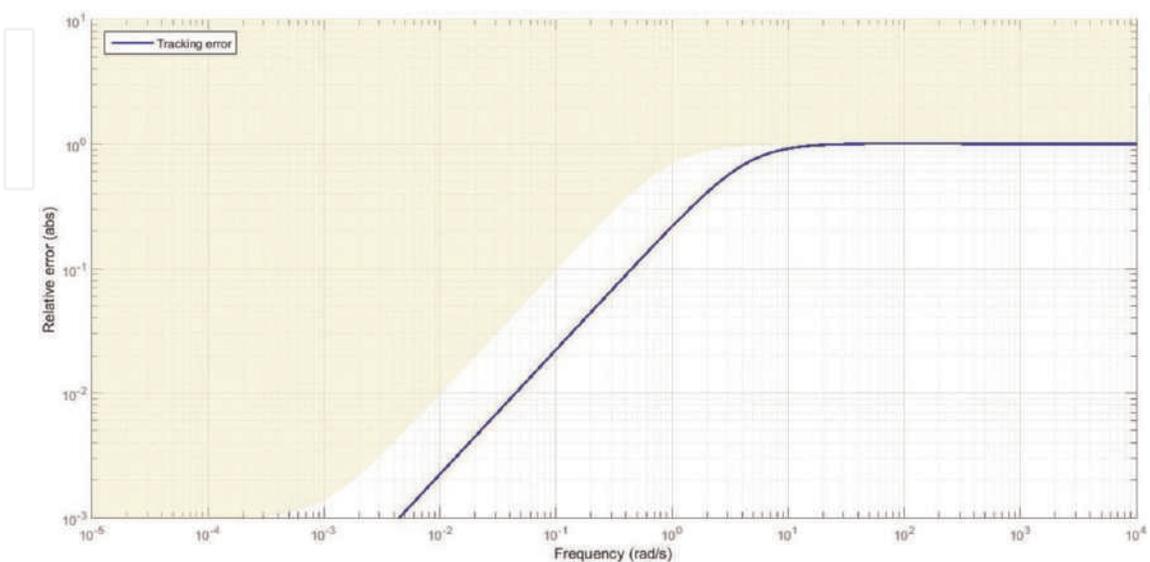


Figure 7.

The performance goal 1: Desired v/s achieved.

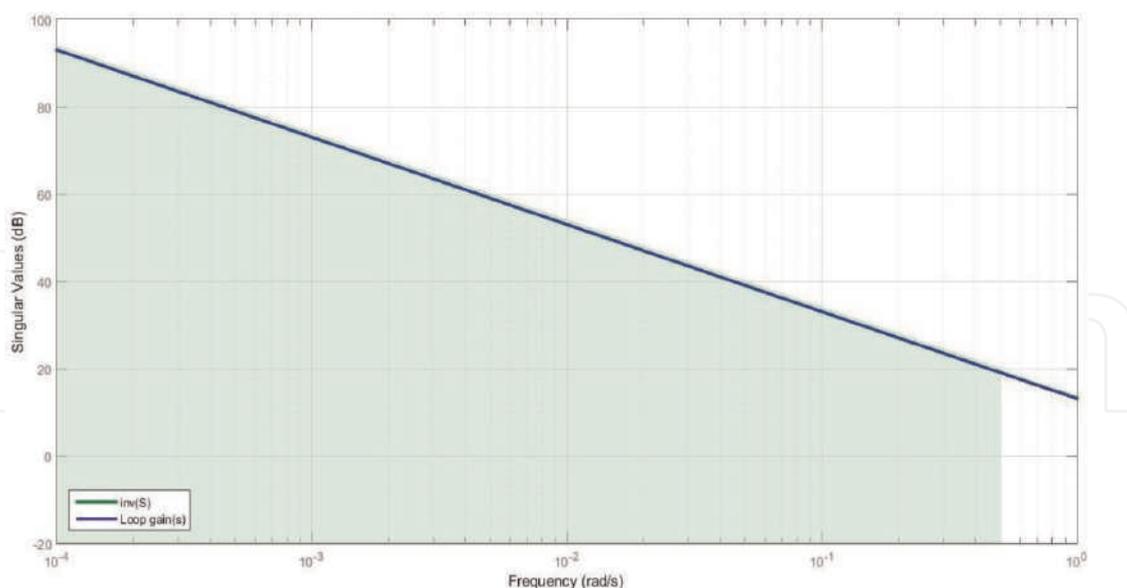


Figure 8.
 The performance goal 2: Desired v/s achieved.

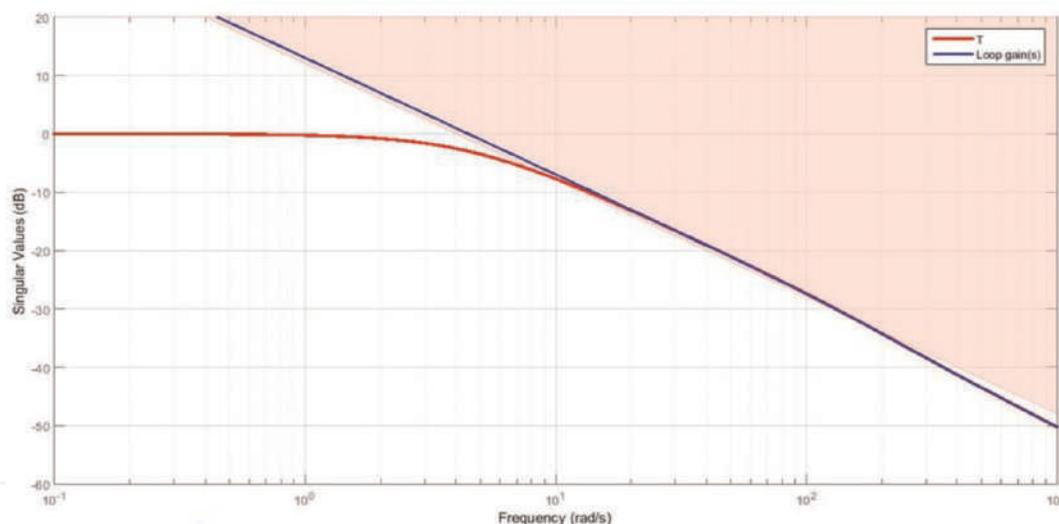


Figure 9.
 The performance goal 3: Desired v/s achieved.

accomplished over all ranges of frequencies below the specified margin. Correspondingly, the max-LG and achieved min-loop v/s ideal values are plotted in **Figures 8 and 9**.

Figure 8 shows the achieved transient duration of proposed model is much less than that desired transient duration. **Figures 10 and 11** demonstrate tracking performance of the proposed model for any arbitrary input. In all cases, the set point of tracking can be accomplished well within the prescribed 2 seconds limitations in system response. From these plots, we infer that simulation of the proposed model guarantees motor internal parameters to remain within tolerable limits of variations for any applied arbitrary external disturbances. Further, the proposed model is tested for simulation with 30 different randomly chosen external disturbances.

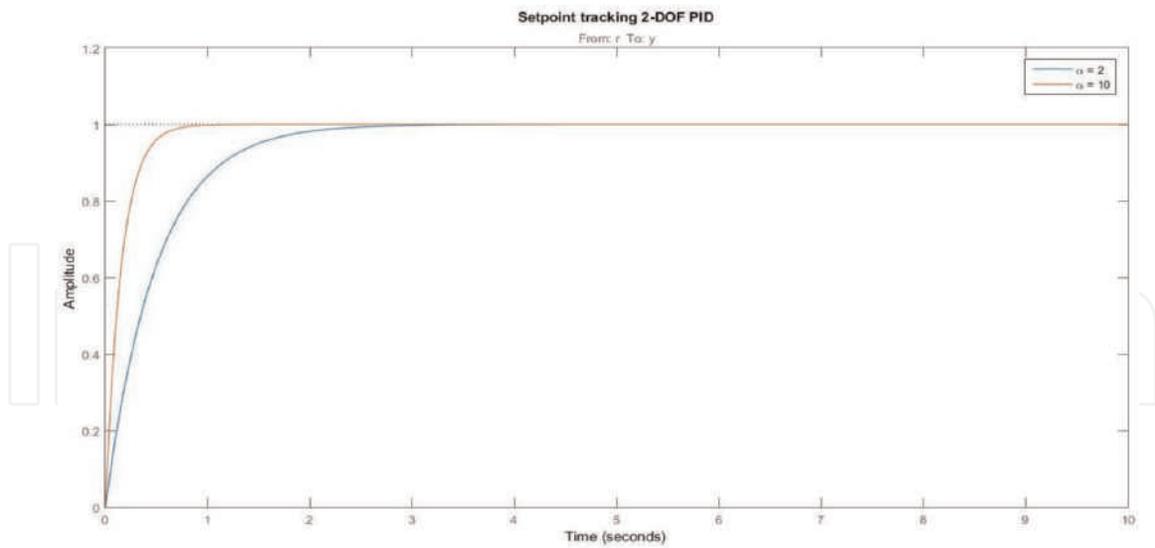


Figure 10.
The performance of tracking with 2-DOF PID controller.

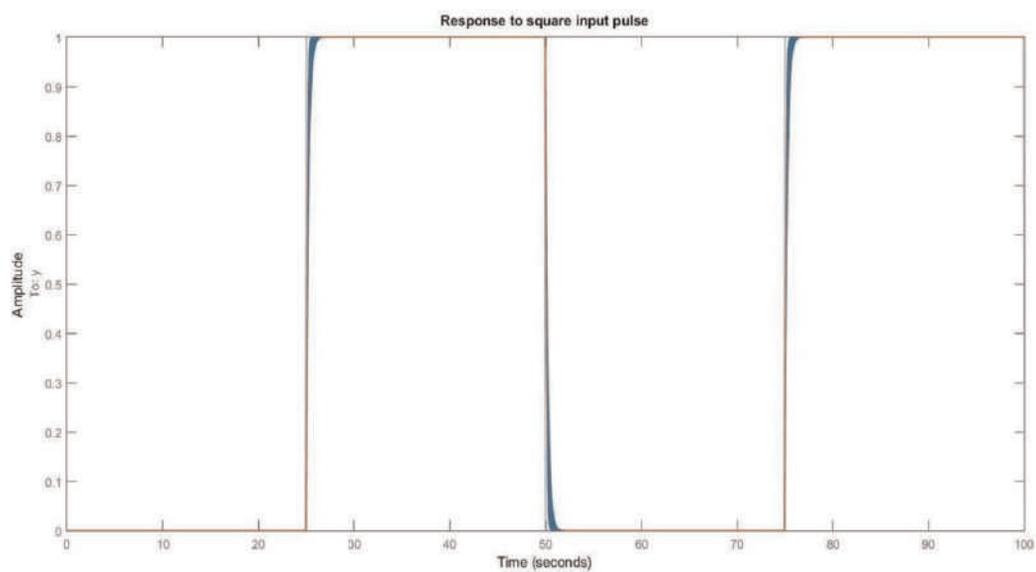


Figure 11.
The performance of speed control to the pulse command (tracking better than the 2 sec).

It is significant to note that achieved performances are far superior to the focused objectives. Additionally, H_{∞} advancement of targeted goals outperforms over expected performance. Out of several performance goals specified for PMDC motors, our work presents simulation results for only three objectives. However, other goals remain to be explored.

Figure 12 represents the disturbance rejection achieved for two different values of proportional and derivative coefficients. The response for randomly chosen load disturbances (30 arbitrary signals) with implicit effect on internal parameters is presented in **Figure 13**. It can be observed that the response for the proposed robust model for controller is well within the accepted bounds. It is worth to observe that the disturbances attenuate well below the prescribed limit of 2 sec.

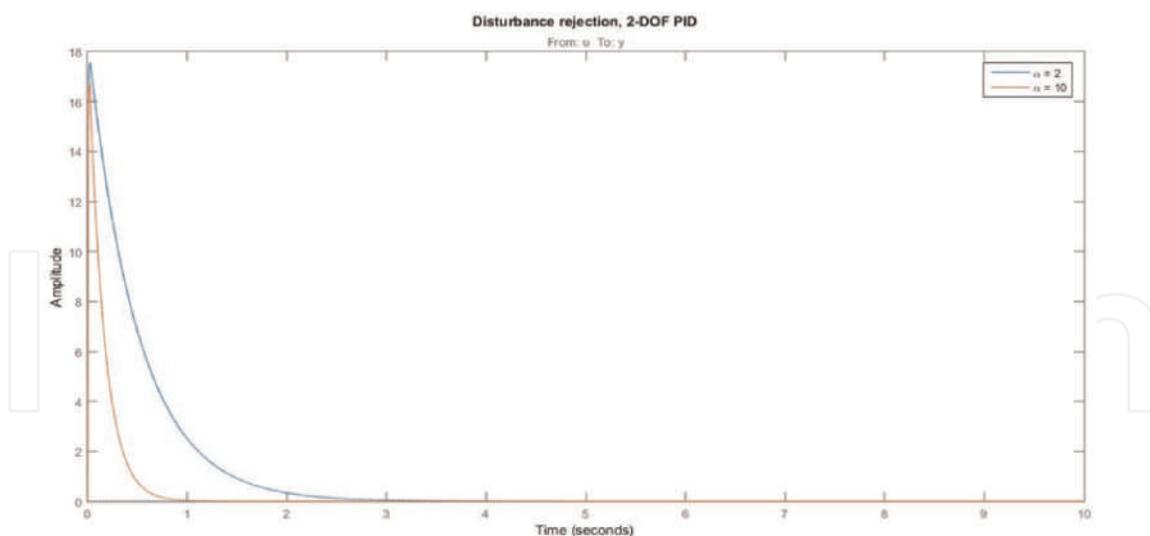


Figure 12.
Disturbance rejection with the controller of 2-DOF PID.

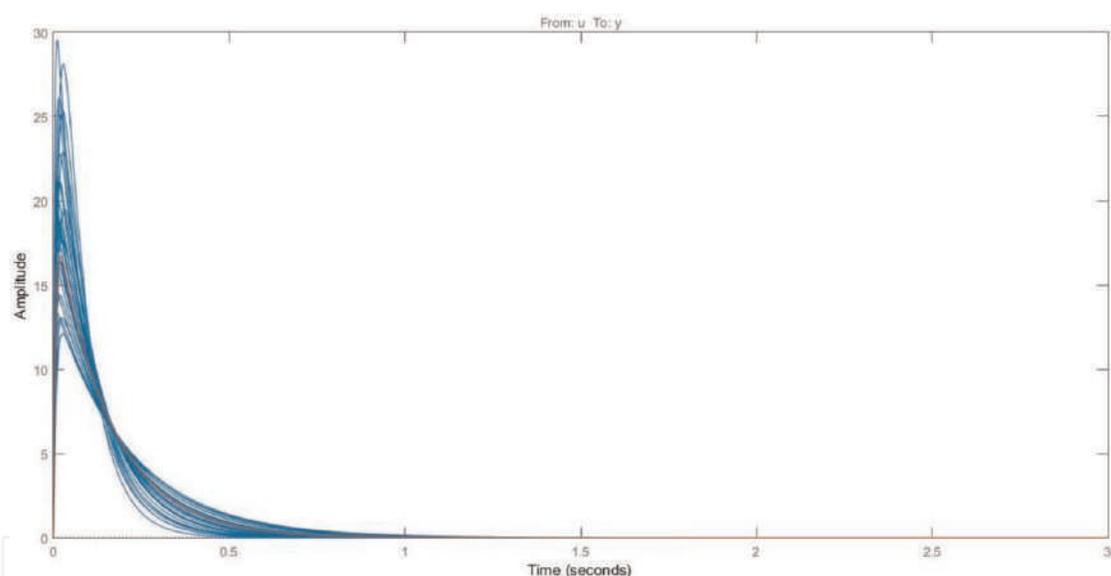


Figure 13.
The response to 30 random disturbances with implicit parameter variations.

7. Conclusion

This work introduces the MO-optimization (Multi Objective) with three performance goals in the structure of H_{∞} for tuning parameters of PID controller for PMDC motor. The two simultaneous competing needs of disturbance rejection along with input tracking are achieved with Robust Controller design with 2-DOF over disturbance rejection using H_{∞} framework. Hence, it is established that the PID controller can be tuned, and corresponding gains can be achieved for a specified constraints/performance capability. Further, H_{∞} provides a better and acceptable framework for optimization.

IntechOpen

Author details

Prasanth Venktareddy^{1*}, Prashanth Narayanappa Anand² and
Prakasha Pundareekane Kanchappa³

1 Nitte Meenakshi Institute of Technology, Bangalore, India

2 BMS Institute of Technology and Management, Bangalore, India

3 Don Bosco Institute of Technology, Bangalore, India

*Address all correspondence to: prasanth.v@nmit.ac.in

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Soni R, Singh D, Pandey P, Sharma P. Simulation of optimal speed control for a DC motor using conventional PID controller and fuzzy logic controller. *International Journal of Information and Computation Technology*. 2013;**3**: 181-188
- [2] Eker I. Sliding mode control with PID sliding surface and experimental application to an electromechanical plant. *ISA Transactions*. 2006;**45**(1): 109-118. DOI: 10.1016/S0019-0578(07)60070-6
- [3] Mamani G, Becedas J, Batlle VF. Robust position control of a DC motor by sliding mode. *IFIP Advances in Information and Communication Technology*. 2010;**314**:495-504. DOI: 10.1007/978-3-642-11628-5_55
- [4] Corradini ML, Ippoliti G, Longhi S, Orlando G. A quasi-sliding mode approach for robust control and speed estimation of PM synchronous motors. *IEEE Transactions on Industrial Electronics*. 2012;**59**(2):1096-1104. DOI: 10.1109/TIE.2011.2158035
- [5] Pal P, Dey R, Pal P, Biswas RK, Bhakta S. Optimal PID controller Design for Speed Control of a separately excited DC motor: A firefly based optimization approach. *International Journal of Soft Computing, Mathematics and Control*. 2015;**4**(4):39-48. DOI: 10.2139/ssrn.3517152
- [6] He J-B, Wang Q-G, Lee T-H. PI/PID controller tuning via LQR approach. *Chemical Engineering Science*. 2000;**55**: 2429-2439. DOI: 10.1016/S0009-2509(99)00512-6
- [7] Nasri M, Nezamabadi-pour H, Maghfoori M. A PSO-based optimum design of PID controller for a linear brushless DC motor. *Proceedings of World Academy of Science, Engineering And Technology*. 2007;**20**:2
- [8] Bautista-Quintero R, Pont MJ. Implementation of H-infinity control algorithms for sensor-constrained mechatronic systems using low-cost microcontrollers. *IEEE Transactions on Industrial Informatics*. 2008;**4**(3): 175-184. DOI: 10.1109/TII.2008.2002703
- [9] Premkumar M, Ram JVR, Jazadamba KPVPB, Rammohan K. Design and simulation of ARTD controller for chemical process industry PMDC motor. *Proceedings of the 4th International Conference on Electrical Energy Systems*. 2018:454-458. DOI: 10.1109/ICEES.2018.8443185

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

171,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Role of Uncertainty in Model Development and Control Design for a Manufacturing Process

Rongfei Li and Francis F. Assadian

Abstract

The use of robotic technology has drastically increased in manufacturing in the twenty-first century. But by utilizing their sensory cues, humans still outperform machines, especially in the micro scale manufacturing, which requires high-precision robot manipulators. These sensory cues naturally compensate for high level of uncertainties that exist in the manufacturing environment. Uncertainties in performing manufacturing tasks may come from measurement noise, model inaccuracy, joint compliance (e.g., elasticity) etc. Although advanced metrology sensors and high-precision microprocessors, which are utilized in nowadays robots, have compensated for many structural and dynamic errors in robot positioning, but a well-designed control algorithm still works as a comparable and cheaper alternative to reduce uncertainties in automated manufacturing. Our work illustrates that a multi-robot control system can reduce various uncertainties to a great amount.

Keywords: uncertainty, modeling, feedback control design, automated manufacturing, robot arm system

1. Introduction

It is believed that the rapid emergence of Robotic technology in industry, and specifically in manufacturing, in the century, will have positive impacts in many aspects of our lives. We have already seen many applications of this technology in macro scale, such as pick and place task [1]. However, there are still applications where humans outperform machines, especially in the micro scale manufacturing, which requires high-precision robot manipulators.

Accurate positioning of robot arms is very important in automated manufacturing field. Over past several decades, we have seen great strides in the technology for accurate positioning robots. We have seen researchers have tried to implement add-on features such as real-time microprocessors, high precision motors, zero backlash gear set, advanced metrology sensors and so on in today's robots. Indeed, they have compensated many structural and dynamic errors in robot positioning [2]. However, those add-on features are usually very expensive and unnecessarily increase the cost during the manufacturing process. Robotic systems that employ a well-designed sensor-based

control strategies can reduce the cost and simultaneously obtain robustness against disturbances and imprecisions from sensors or modeling.

The process of fastening and unfastening a screw is a mundane but a challenging task in the automated manufacturing. We found recent research on this topic only focuses on how robots should generate push/pull force on a driver [3]. The axial forces and torques are first measured through sensors and then controlled to imitate human approach of fastening and unfastening by applying similar amount of axial forces and torques. This approach only considers tactile sensing, however, human beings, also use the information from visual sensing to help with this task. To replicate visual sensing in robots, for example, a camera system could be utilized to make sure a tool is at the right pose (correct orientation and location where head of bolt and tail of driver coincide). A visual system that can provide an accurate and repeatable positional tracking of the tool becomes significantly important and useful not only in this type of an application but also in many other applications of the automated manufacturing [4].

In this work, we have designed a multi-robotic control system that simulates the positioning process for fastening and unfastening applications and have examined its robustness against various uncertainties, which may occur, in this process. This control system is a visual servoing system where a camera is mounted on a robot arm manipulator and provides vision data for the motion control of a second robot manipulator with a tool. Both the Position-Based Visual Servoing (PBVS) and the Image-Based Visual Servoing (IBVS) systems have been thoroughly investigated in [5–8]. However, in this related work, in the visual servoing domain, the development of the outer-loop controller is usually achieved with the PID controller, or its simplified variations based only on a kinematic model of camera [5, 8]. One improvement in this work is to use Youla robust control design technique [9] that includes both kinematics and dynamics in the model development stage. The increase in the model fidelity for the control design can positively influence the precision of the feature estimation and the control system stability for the high-speed tasks. Benefits of our design are discussed in more details in the following sections.

Position control algorithms for both the visual and the tool manipulation systems are discussed in this Chapter. Especially, a combination of a feedforward and a feedback control architecture has been designed for the tool manipulation system, which enables the tool to move fast to a desired location with a high precision in its final pose. Simulation results for the Single Input Single Output (SISO) case in various scenarios are presented and furthermore, the robustness to various noise sources in this manufacturing process are examined.

2. Literature review

We have seen many efforts been made to improve the positioning accuracy of robotic systems in the past few decades. An effective way to reduce the amount of inaccuracy is to measure it with sensors and compensate it through feedback control loop. Many metrology techniques have been investigated and applied for different kinds of data capturing. Among them, three methods have gained the most popularity in the recent research, namely, vision-based methods, tactile-based methods, and vision-tactile integrated methods. In this section, we will briefly review those approaches and their applications.

2.1 Vision-based methods

The vision-based methods have been widely developed in the recent years and used to determine position and orientation of target objects in robotic systems. Zhu et al. discussed Abbe errors in a 2D vision system for robotic drilling. Four laser displacement sensors were used to improve the accuracy of the vision-based measurement system [10]. Liu et al. proposed a visual servoing method for positioning in aircraft digital assembly [11, 12]. With the measurements from two CCD cameras and four distance sensors, the proposed method can accurately align the positioner's ball-socket with the ball-head fixed on the aircraft structures in a finite time.

In addition to mentioned applications, we have seen many contributions to the vision-based methods in robotic manipulation. However, most of those researchers focused on success rate of grasping on end-effector without enough analysis on the positioning accuracy. Du et al. published a study for the robotic grasp detection by visually localizing the object and estimating its pose [13]. Avigal et al. proposed a 6-DoFs grasp planning method using fast 3D reconstruction and grasp quality convolutional neural network (CNN) [14]. Wu et al. proposed an end-to-end solution for visual learning [15].

2.2 Tactile-based methods

In addition, with the development of tactile sensors in the last few years, we have seen more and more focus on tactile-based methods in robotic positioning domain. The tactile sensors can show contact states of the end-effector and the object in robotic manipulations. The contact state can be used to determine objects' relative orientations and positions with respect to the gripper. Li et al. designed a tactile sensor of GelSight and generated tactile maps for different poses of a small object in the gripper [16]. He studied the localization and control manipulation for a specific USB connector insertion task. Dong et al. studied the tactile-based insertion task for dense box packing with two GelSlim fingers which are used to estimate object's pose error based on neural network [17]. Furthermore, Hogan et al. developed a tactile-based feedback loop in order to control a dual-palm robotic system for dexterous manipulations [18]. Those tactile-based methods can only realize relative accurate positioning of the tool with the end-effector, but the positioning of the robot manipulator itself is not addressed.

2.3 Vision-tactile integrated methods

Vision sensing can provide more environment information with a wide measurement range, while tactile sensing can provide more detailed information in robotic manipulations. Therefore, the vision-tactile integrated methods came into being. Fazeli et al. proposed a hierarchical learning method for complex manipulation skills with multisensory fusion in seeing and touching [19]. Gregorio et al. developed a manipulation system for automatic electric wires insertion performed by an industrial robot with a camera and tactile sensors implemented on a commercial gripper [20].

According to the analysis, all the integrated sensory applications have achieved accurate robotic manipulation tasks such as insertion and their performances have been verified in experiments. However, the error space in those references is usually

small and none of them has considered all the translational and rotational errors in 6 DoFs. Moreover, tactile-based or vision-tactile integrated methods will increase expense of massive manufacturing because tactile sensors are more expensive to purchase and maintain compared to visual sensors. Based on that, our work is to explore the capability of the vision-based methods and design a new method to improve the accuracy of positioning in the robot manipulation system.

3. Technical background

In the first part of this section, we will briefly discuss the algorithm of the traditional Image based visual servoing (IBVS) and its drawbacks. In the second part, we will review sources of all possible uncertainties occurred in automated manufacturing and existing approaches for reducing them.

3.1 A brief overview of the classical IBVS architecture

The control of the visual system, the eye-in-hand camera configuration [21], is discussed in this section. The general visual servoing problem can be divided into two categories: PBVS and IBVS [5]. This work focuses on the IBVS structure. **Figure 1** shows the control block diagram for a classical IBVS architecture. In **Figure 1**, $s = [u, v]^T$ is the image feature position vector, $s^* = [u^*, v^*]^T$ is the target image feature position vector, and their difference $e = s^* - s$ is the error vector. The IBVS structure is a cascaded control loop with an outer controller and an inner joint controller. The outer controller takes the feature error e as input and generates the positional targets, which are denoted as \dot{q} in the **Figure 1**. The inner controller

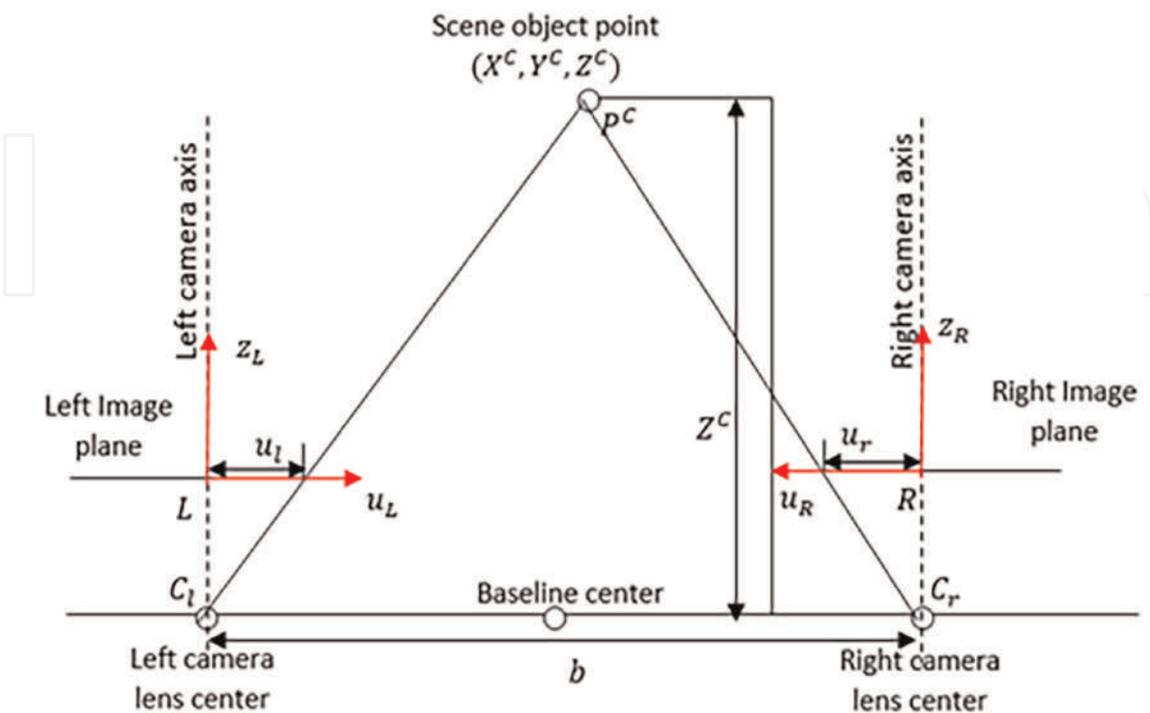


Figure 1.
The block diagram of the classical IBVS control architecture.

stabilizes the joints to generated positional targets from outer controller. L_e is the so-called interaction matrix [5], which is a 2-by-6 matrix, and relates the time derivative of the image feature s to the spatial velocity of the camera V_c , a column vector of six-elements, by the following:

$$\dot{s} = L_e V_c \quad (1)$$

We can design a proportional controller to force the error to exponentially converge to zero, i.e.:

$$\dot{e} = -ke, \quad k > 0 \quad (2)$$

Suppose the target image feature is a constant; that is $\dot{s}^* = 0$, hence, we can derive from Eq. (2):

$$\dot{e} = \dot{s}^* - \dot{s} = -L_e V_c \quad (3)$$

From Eqs. (2) and (3), one will be able to obtain:

$$V_c = kL_e^+ e \quad (4)$$

where L_e^+ is the pseudoinverse of L_e . The derivation of the interaction matrix with a monocular camera is further explained next. We assume a point with the three-dimension (3D) coordinates in the camera frame is given as $P^C = [X^C, Y^C, Z^C]$. We further assume a zero-skew coefficient, i.e., $s_C = \mathbf{0}$ in Eq. (2) with a baseline distance $b = \mathbf{0}$, for a monocular camera model, then the image feature coordinate $s = [u, v]^T$ can be expressed as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{f_u X^C}{Z^C} + u_0 \\ \frac{f_v Y^C}{Z^C} + v_0 \end{bmatrix} \quad (5)$$

Taking the time derivative of Eq. (5), we can obtain:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f_u (\dot{X}^C Z^C - \dot{Z}^C X^C)}{(Z^C)^2} \\ \frac{f_v (\dot{Y}^C Z^C - \dot{Z}^C Y^C)}{(Z^C)^2} \end{bmatrix} \quad (6)$$

The rigid body motion of a 3D point in the camera model can be derived as:

$$\dot{P}^C = v^C + \omega^C \times P^C \Leftrightarrow \begin{cases} \dot{X}^C = v_X^C - \omega_Y^C Z^C + \omega_Z^C Y^C \\ \dot{Y}^C = v_Y^C - \omega_Z^C X^C + \omega_X^C Z^C \\ \dot{Z}^C = v_Z^C - \omega_X^C Y^C + \omega_Y^C X^C \end{cases} \quad (7)$$

Substituting (7) into (6), and rearranging the terms, we obtain:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{f_u}{Z^C} & 0 & -\frac{u-u_0}{Z^C} & -\frac{(u-u_0)(v-v_0)}{f_v} & \frac{f_u^2 + (u-u_0)^2}{f_u} & -\frac{f_u(v-v_0)}{f_v} \\ 0 & \frac{f_v}{Z^C} & -\frac{v-v_0}{Z^C} & -\frac{f_v^2 + (v-v_0)^2}{f_v} & \frac{(u-u_0)(v-v_0)}{f_u} & \frac{f_v(u-u_0)}{f_u} \end{bmatrix} \begin{bmatrix} v_X^C \\ v_Y^C \\ v_Z^C \\ \omega_X^C \\ \omega_Y^C \\ \omega_Z^C \end{bmatrix} \quad (8)$$

Eq. (8) can be simply written as:

$$\dot{s} = L_e V_c = L_e \begin{bmatrix} v^C \\ \omega^C \end{bmatrix} \quad (9)$$

Some drawbacks of the classical IBVS are summarized next. To compute the interaction matrix L_e from Eq. (8), the depth Z^C needs to be estimated. This can be usually approximated as either the depth of the initial position or the depth of the target position or their average value [5]. A careless estimation of the depth may lead to a system instability. In addition, the design of the proportional controller is based on, Eq. (1), the camera kinematic relationships, such that there is no dynamics considered in this model. The kinematic model is sufficient for very slow responding system, however, for faster responses, one has to take into account the manipulator dynamics along with the camera model.

In this work, we propose a new controller algorithm, similar to the classical IBVS structure, where the controller is designed with the complete dynamic and kinematic models of the robot manipulator and the camera. Furthermore, this algorithm does not require any depth estimation, therefore, it will not be necessary to use the interaction matrix. The development of this new algorithm is presented in sections 6 and 7 of this Chapter.

3.2 A brief overview of sources of uncertainties and approaches for reduction

Uncertainties in automated manufacturing can originate from different sources. We can divide these uncertainties into two categories: sensor measurement noise, and dynamic and kinematic modeling errors from both the measurement system and the robot manipulators. This section briefly reviews each uncertainty source including the proposed methods for reducing these uncertainties.

3.2.1 A brief overview of a stereo camera model and its calibration

A camera model (i.e., the pin-hole model [22]) has been adopted in the visual servoing techniques to generate an interaction matrix [5]. The object depth, the distance between a point on the object and the camera center as illustrated in **Figure 2**, needs to be either estimated or approximated by an interaction matrix [5]. One of the methods is to directly measure the depth by a stereo (binocular) camera with the use of two image planes [23].

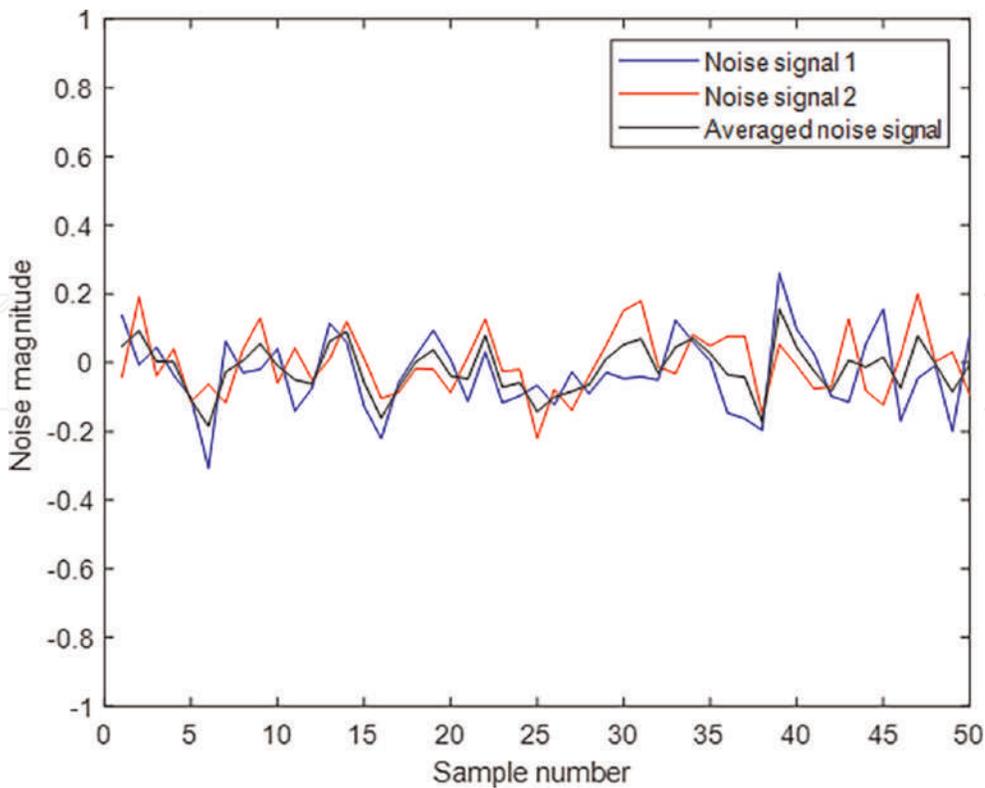


Figure 2.
 The projection of a scene object on the stereo camera's image planes.

As shown in **Figure 2**, two identical cameras are separated by a baseline distance b . An object point, $\mathbf{P}^C = [X^C, Y^C, Z^C]^T$, which is measured in the camera frame, at the baseline center, is projected to two parallel virtual image planes, and each plane is located between each optical center (C_L or C_R) and the object point \mathbf{P}^C . The intrinsic camera parameters relate the coordinates of the object point in the camera frame and its corresponding image coordinates $\mathbf{p} = (u, v)$ on each of the image plane with an exact mathematical relationship. This relationship is given by:

Note: v coordinate on each image plane is not shown in the plot but is measured along the axis that is perpendicular to and point out of the plot.

$$\begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \frac{1}{Z^C} \begin{bmatrix} f_u & s_c & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} - \frac{b}{2Z^C} \begin{bmatrix} f_u \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \frac{1}{Z^C} \begin{bmatrix} f_u & s_c & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^C \\ Y^C \\ Z^C \end{bmatrix} + \frac{b}{2Z^C} \begin{bmatrix} f_u \\ 0 \\ 0 \end{bmatrix} \quad (11)$$

Where, f_u and f_v are the horizontal and the vertical focal lengths, and, s_c is a skew coefficient. In most cases, f_u and f_v are different if the image horizontal and vertical axes are not perpendicular. In order not to have negative pixel coordinates, the origin of the image plane will be usually chosen at the upper left corner instead of the center. u_0 and v_0 describe the coordinate offsets. The camera model uncertainties may arise from the estimation of those camera intrinsic parameter values. The camera calibration can be used to precisely estimate these values.

The stereo camera calibration has been well studied in [24–27]. As summarized in [24], the calibration method can be divided into two broad categories: the photogrammetric calibration and the self-calibration. In the photogrammetric calibration [25], the camera is calibrated by observing a calibration object whose geometry is well known in the 3D space. These methods are very accurate but require an expensive apparatus and elaborate setups [24]. The self-calibration [24, 26, 27] is performed by finding the equivalences between the captured images of a static scene from different perspectives. Although cheap and flexible, these methods are not always reliable [24]. The author in [24] proposed a new self-calibration technique that observe planar pattern at different orientations and showed improved results.

3.2.2 A brief overview of the robot manipulator model and its calibration

In this work, we consider the elbow manipulators [28] with the spherical wrist in the multi-robot system to move an end-effector freely in 6 degrees of freedoms (dofs). This model of the robot has six links with three for the arms and the other three for the wrist. The robot arms freely move the end effector to any position in the reachable space with 3 dofs while the robot spherical wrists allow the end effector to orient in any directions with another 3 dofs. For the elbow manipulators, a joint is connected between each two adjacent links and there are in total six convolutional joints. The specific industry model of this type is ABB IRB 4600 [29].

A commonly used convention for selecting and generating the reference frames in the robotic applications is the Denavit-Hartenberg convention (or D-H convention) [30]. Suppose each link is attached to a Cartesian coordinate frame, $O_i X_i Y_i Z_i$. In this convention, each homogeneous transformation matrix A_i (from frame $i - 1$ to frame i) can be represented as a product of four basic transformations:

$$A_i = Rot_{z,q_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} = \begin{bmatrix} c_{q_i} & -s_{q_i}c_{\alpha_i} & s_{q_i}s_{\alpha_i} & a_i c_{q_i} \\ s_{q_i} & c_{q_i}c_{\alpha_i} & -c_{q_i}s_{\alpha_i} & a_i s_{q_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Note: $c_{q_i} \equiv \cos(q_i)$, $c_{\alpha_i} \equiv \cos(\alpha_i)$, $s_{q_i} \equiv \sin(q_i)$, $s_{\alpha_i} \equiv \sin(\alpha_i)$.

Where q_i , a_i , α_i and d_i are parameters of link i and joint i , a_i is the link length, q_i is the rotational angle, α_i is the twist angle and d_i is the offset length between the previous $(i - 1)^{th}$ and the current i^{th} robot links. The quantities of each parameter in (12) are calculated based on the steps in [28].

We can generate the transformation matrix from the base frame $O_0 X_0 Y_0 Z_0 (P^0)$ to the end-effector frame $O_6 X_6 Y_6 Z_6 (P^6)$:

$$T_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 \quad (13)$$

If any point with respect to the end effector frame P^6 is known, we can calculate its coordinate with respect to the base frame P^0 as:

$$P^0 = T_6^0 P^6 \quad (14)$$

In addition, the transformation from the base frame P^0 to the end-effector frame P^6 can be derived:

$$T_0^6 = (T_6^0)^{-1} \quad (15)$$

which is used to generate the image coordinates of a point captured by a camera with its center attached to the end effector, from the 3D coordinates of a point in the base frame.

Eq. (13) shows that the position of the end-effector P^{end} (where P^{end} is the origin of the end-effector frame P^6) is a function of all the joint angles $q = [q_i | i \in 1, 2, 3, 4, 5, 6]$ and the parameters $Pa = [a_i, \alpha_i, d_i | i \in 1, 2, 3, 4, 5, 6]$:

$$P^{end} = \mathcal{F}(q, Pa) \quad (16)$$

Eq. (16) describes the forward kinematic model of the robot manipulator, which could be utilized to calculate the position of the end effector from the joint angles and the robot parameters. The inverse process is called the inverse kinematic, which the joint angles can be computed from the position and the parameters. The estimation of the robot parameters Pa determines the accuracy of the kinematic models of the robot manipulators.

The paper [31] provides a good summary of the current robot calibration methods. The author of [32] states that over 90% of the position errors are due to the errors in the robot zero position (the kinematic parameter errors). As a result, most researchers focus on the kinematic robot calibration (or level 2 calibration [31]) to enhance the robot absolute positioning accuracy [33–36]. Generally, the kinematic model-based calibration involves four sequential steps: Modeling, Measurement, Identification, Correction. Modeling is a development of a mathematical model of the geometry and the robot motion. The most popular one is D-H convention [30] and other alternatives include S-model [37] and zero-reference model [38]. At the measurement step, the absolute position of the end-effector is measured from the sensors, e.g., the acoustic sensors [37], the visual sensors [34], etc. In the identification step, the parameter errors for the robot are identified by minimizing the residual position errors with different techniques [39, 40]. This final step is to implement the new model with the corrected parameters.

On the other hand, the non-kinematic calibration modeling (level 3 calibration [31]) [39, 41], which includes the dynamic factors such as the joint and the link flexibility in the calibration, increase accuracy of the robot calibration, but complicates the mathematical functions that govern the parameters relationship.

3.2.3 The image averaging techniques for denoising

The image noises are inevitably introduced in the image processing. Several image denoising techniques have been proposed so far. A good noise removal algorithm ought to remove as much noise as possible while safeguarding the edges. The Gaussian white noise has been dealt with using the spatial filters, e.g., the Gaussian filter, the Mean filter and the Wiener filter [42]. The noise reduction using the wavelet methods [43, 44] have benefits of keeping more useful details but at the expense of the computational complexity. However, depending on the selected wavelet methods, the filters that operate in the wavelet domain still filter out (or blur) some important high frequency useful information of the original image, even though more edges are preserved with the wavelet method when comparing with the spatial filter approaches.

All the aforementioned methods present ways to reduce noise in the image processing starting from a noisy image. We can approach this problem with the multiple noisy images taken from the same perspective. Assuming the same perspective ensures the same environmental conditions (illumination, temperature, etc.) that affect the image noise level. Given the same conditions, an image noise level taken at a particular time should be very similar to another image taken at a different time. This redundancy can be used for the purpose of improving image precision estimation in the presence of noise. The method that uses this redundancy to reduce noise is called signal averaging (or the image averaging in the application of the image processing) [45]. The image averaging has a natural advantage of retaining all the image details as well as reducing the unwanted noises, given that all the images for the averaging technique are taken from the same perspective. The robot's rigid end effector that holds a camera minimizes shaking and drift when shooting pictures. Furthermore, in the denoising process, the precise estimations require that the original image details to be retained. Considering these previous statements, we decided to choose image averaging over all other denoising techniques in this work.

The image averaging technique is illustrated in **Figure 3**. Assume a random, unbiased noise signal, and in addition, assume that this noise signal is completely uncorrelated with the image signal itself. As noisy images are averaged, the original true image is kept the same and the magnitude of the noise signal is compressed thus improving the signal-to-noise ratio. In **Figure 3**, we generated two random signals with the same standard deviation, and they are respectively represented by the blue and the red lines. The black line is the average of the two signals, whose magnitude is significantly decreased compared to each of the original signal. In general, we can come up with a mathematical relationship between the noise level reduction and the sample size for averaging. Assume we have N numbers of Gaussian white noise samples with the standard deviation σ . Each sample is denoted as z_i , where i represents i^{th} sample signal. Therefore, we can acquire that:

$$var(z_i) = E(z_i^2) = \sigma \quad (17)$$

where $E(\cdot)$ is the expectation value and σ is the standard deviation of the noise signal. By averaging the N Gaussian white noise signals, we can write:

$$var(z_{avg}) = var\left(\frac{1}{N} \sum_{i=1}^N z_i\right) = \frac{1}{N^2} N \sigma^2 = \frac{1}{N} \sigma^2 = \left(\frac{1}{\sqrt{N}} \sigma\right)^2 \quad (18)$$

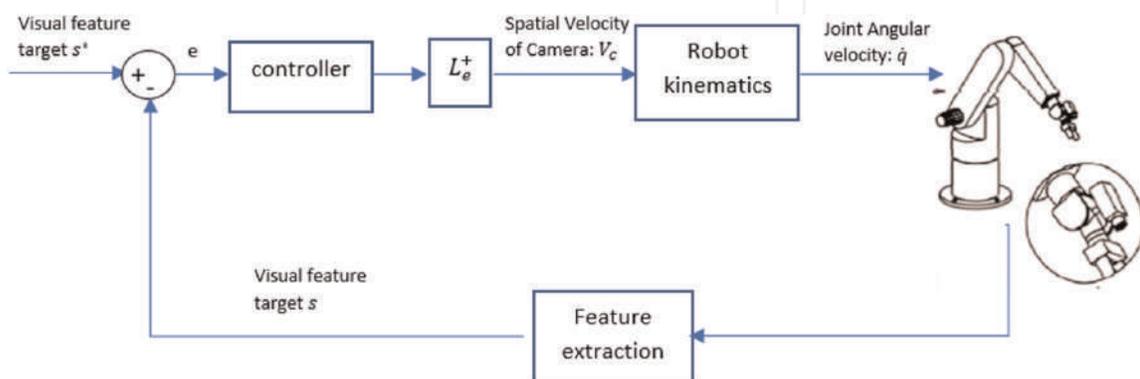


Figure 3.
An example of a noise level reduction by image averaging.

where \mathbf{z}_{avg} is the average of the N noise signals. Eq. (18) demonstrates that a total number of N samples are required to reduce the signal noise level by \sqrt{N} . Since our goal is to reduce the image noise to within a fixed threshold (a constant number expressed as a standard deviation), a smaller variation in the original image requires much less samples to make an equivalent noise-reduction estimation. Thus, it is worthwhile for the camera to move around, rather than being stationary, in order to find the best locations where the image noise level estimation is small. In general, we can reduce the noise level as much as needed by taking more samples.

3.2.4 The dynamic errors and its modeling in the feedback control loop

The dynamic errors in a robot manipulator consist of any joint non-static errors. Among these error sources, which have the most significant effect on the robot position control accuracy, are the deviations between the actual joint rotation with its measured value from the unmodeled dynamic uncertainties, such as backlash, friction, compliance due to gears' elimination, joint or link flexibility and thermal effects. As discussed in Section 3.2.2, we can account for all these errors in the dynamic modeling step by developing a high-fidelity dynamic model, where all these parameter values could be identified through calibration. An easier way is to regard all these dynamic errors as disturbances to a manipulator control system. The control system is able to make the plant output track the desired input (the reference signal) and while simultaneously, it rejects these disturbances. The design of the robot manipulators control systems and the demonstrations of the capability of these feedback loops to reject these aforementioned dynamic errors are discussed in more detail in the later sections.

4. The topology of the multi-robotic system for accurate positioning control

In this section, we discuss the proposed control architectures for a multi-robot system, which enables the high-accuracy movement of a tool in various manufacturing scenarios by reducing the process uncertainties. Assuming at the start, all camera and robot manipulators are well calibrated by using one or multiple methods discussed in Sections 3.2.1 and 3.2.2, so that the initial camera and robot manipulators parameters are identified. Therefore, in this case, the main uncertainties include the sensor noise and the dynamic modeling errors. **Figure 4** shows the overall topology of this multi-robot system.

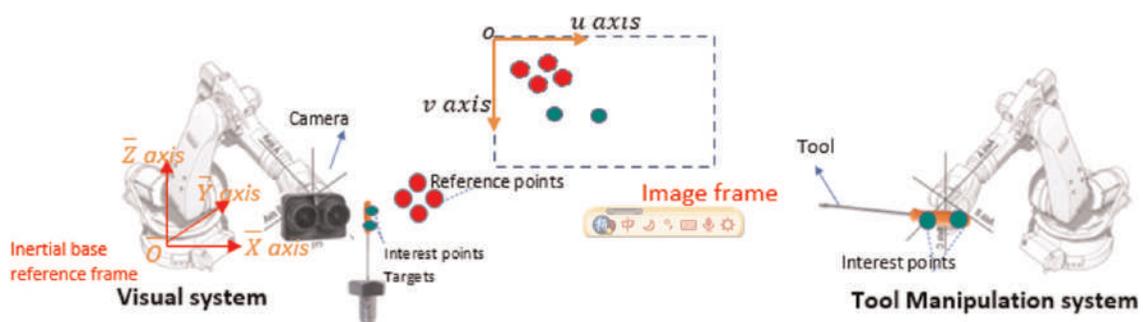


Figure 4.
 The topology of the multi-robotic system for accurate positioning control.

The multi-robot system is composed of a visual system and a tool manipulation system (**Figure 4**). In the visual system, a camera is mounted on an elbow robot arm while a tool is held by the end-effector of the robot manipulator arm. The goal of the visual system is to provide precise estimation of the tool pose so that the tool manipulator can control the pose with the guidance from the visual system. Two fiducial markers (green circles or the interest points) are placed on the tool to help the computer to detect the position and the orientation of the tool. The absolute coordinates of the reference points (red circles) are known in the inertial reference frame. The reference points are placed close to the tool's target location (the target interest points) so that the reference points and the target interest points can be captured in the camera frame when the tool gets close to its target pose.

Four reference points are selected close to each other in the space. In a visual servoing problem, a location in space from which an image was taken can only be uniquely determined by at least four points, 3 points to determine a specific location and one point to determine the orientation. This is a location determination problem (LDP) using image recognition [46]. Therefore, we consider using four reference points to determine the camera pose in the 3D space. However, whenever the camera pose is fixed and known in space, the stereo camera, which can detect the depth, provides the distinct 3D location of a point from the image coordinates.

4.1 The multi-robotic system sequential control procedure

The movement control of the robot manipulators is asynchronous in the visual and the tool manipulator systems. A flowchart demonstrating this sequential control process is shown in **Figure 5**.

The first stage consists of the optimal camera pose determination and control. In this stage, the camera moves and searches for an optimal position based on the minimization of a proposed objective function, in this case, the time duration and the energy consumption, while reducing the image noise of the reference points to within an acceptable threshold. In the second stage, the camera movement adjustment control, any uncertainties occur in the movement of camera from the last stage is eliminated by an eye-in-hand visual servoing controller. After the movement adjustment, the camera is kept static and provides precise estimations of the tool position. In the last stage, the high-accuracy tool manipulation control, the tool movement is controlled and guided to the target pose location by an eye-to-hand visual servoing controller. Each control method and architecture are discussed in the sections below.

4.2 The optimal camera pose determination process and its control architecture

Figure 6 shows how the optimal pose of the camera is determined from a single picture taken at different perspectives. The uncertainty in the image processing is spatially related. As the camera moves in space, the combined factors (the light conditions, the temperature, etc.) that affect the image processing changes and with



Figure 5.
The flow chart of the sequential control procedure.

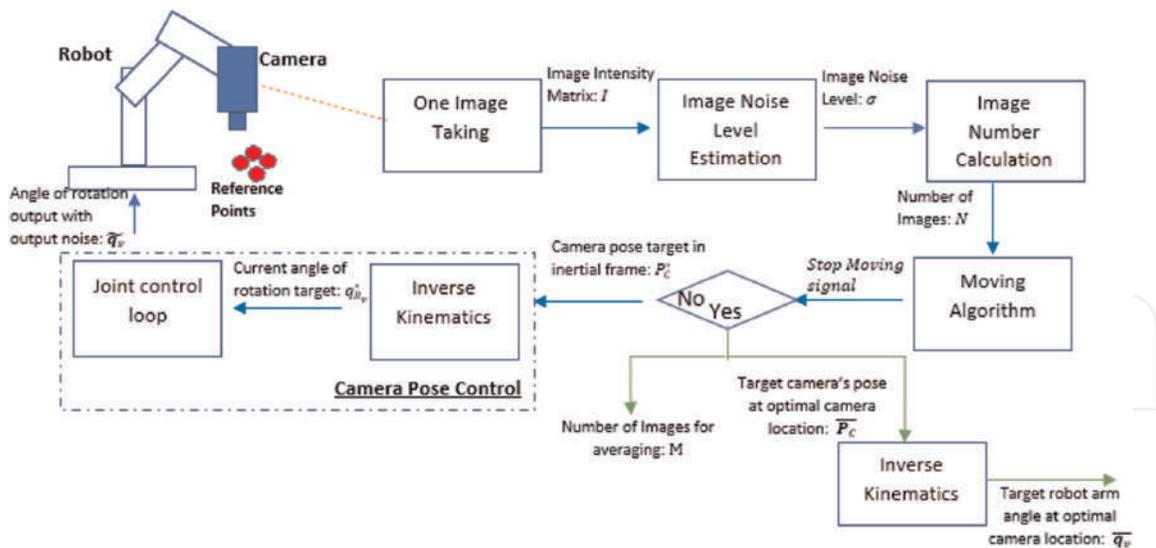


Figure 6.
 The optimal camera pose determination process and its control architecture.

these changes, the uncertainty level in the estimation changes accordingly. In this work, we propose to apply image averaging [45] to reduce the uncertainty level in the pose estimation. As discussed in Section 3.2.3, the number of images required for the averaging increases by a factor of 2 for reducing the uncertainty level by the square root of the same factor. In order to reduce the energy consumption and the time duration in this photo taking process, it is necessary to first determine the location where the image averaging should take place before the camera actually starts to take multiple photos.

In **Figure 6**, in the first stage, the camera takes a single picture. In the second stage, we compute the image intensity matrix I from that photo and then, we estimate the noise level σ across the image by a previously developed algorithm, see [47]. In the third stage, we calculate the uncertainty level from the image noise level and generates the number of images N required to reduce this uncertainty within a prescribed threshold. In the fourth stage, utilizing a moving algorithm, which is designed as a part of this work, the current camera target pose, P_C is commanded. In the fifth stage, the camera pose controller guides the camera to the target pose location using the encoders that measure joints rotational angles, \tilde{q}_v . These five stages are repeated until the movement algorithm instructs the camera to stay in the current pose. Then, this current target pose is the optimal pose \overline{P}_C of the camera where the total energy consumption and the time duration is minimized. The output \overline{q}_v is the target joint angles of visual manipulator system at the optimal pose of the camera. If for any reason, such as uncompensated uncertainties, the current pose is not the same as the optimal pose, \overline{P}_C , then the camera movement adjustment control, presented in the next section, will reduce this error. In addition, M is the number of pictures needed for the averaging at the optimal pose location of the camera.

4.3 The camera movement adjustment control block diagram

We propose a control method with its associated block diagram for the camera movement adjustment as shown in **Figure 7**. The role of this feedback control is to deal with the errors occurred in the dynamics and the measurements of the previous stage.

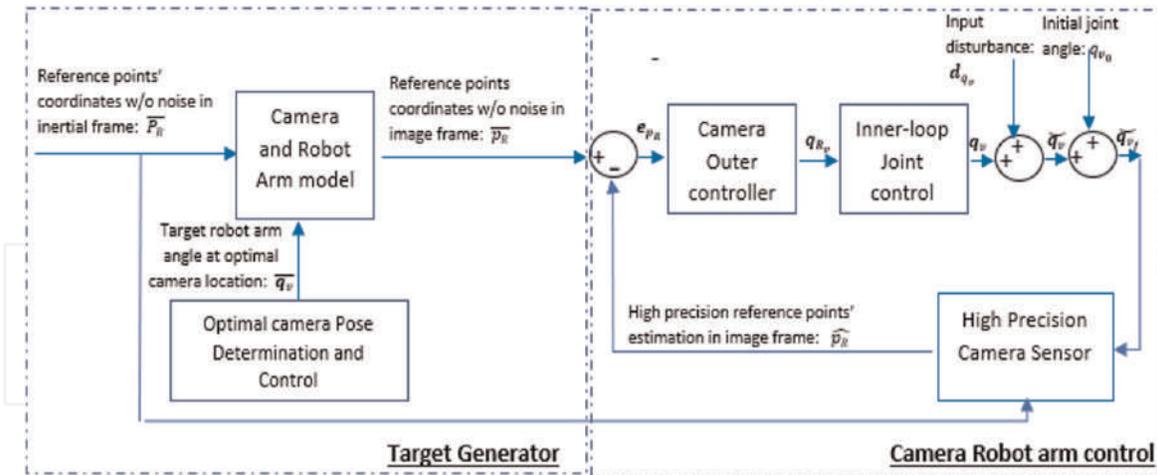


Figure 7.
The camera movement adjustment control block diagram.

In **Figure 4**, four reference points whose absolute positions are known in the space, are selected close to the tool target pose. The fiducial markers are placed on the reference points, so that their location can be recognized and estimated in a 2D image coordinate frame using computer vision. From the kinematic model of the robot arm and the camera, the image coordinates of the reference points can be calculated online, and those coordinates are used as the targets for a cascaded control loop and are noted as \bar{p}_R in **Figure 7**. After applying the image averaging technique (Section 3.2.3), we can obtain a precise estimation of the current position of the reference points and are noted as \hat{p}_R in **Figure 7** in image coordinates from the computer vision. Therefore, any deviation between \bar{p}_R and \hat{p}_R could be the result of some uncertainties, such as the joint compliances, which are not compensated by the joint control loop as shown in **Figure 6**. The cascaded controller is similar to the image-based visual servoing scheme (IBVS), as discussed in Section 3.1. The inner-loop control strategy in this part is also very similar to the joint control in the camera pose control in **Figure 6**, and its control design and simulation results are discussed in Section 5.

4.4 The high-accuracy tool manipulation control block diagram

We propose a control strategy with its associated control block diagram for the tool manipulation system as shown in **Figure 8**. The control algorithm in this block diagram is a combination of a feedforward and a feedback control.

The feedforward control loop is an open loop which brings the tool as close to the target position as possible in the presence of the input disturbance, d_{q_m} . In the inner joint control loop, the noise sources may originate from the low fidelity cheap encoder joint sensors and the dynamic errors from the joint, e.g., compliances. All sources of noise from the joint control loop are combined and modeled as an input disturbance, d_{q_m} , to the outer control loop. The outputs of the feedforward are the reference joint angles of rotations, $q_{R_m}^{feedforward}$, which are added to the outer feedback controller outputs, $q_{R_m}^{feedback}$, and set as the targets for the joint control inner-loop. The function of the forward kinematics is to transform a set of current joint angles of the tool manipulator to the current pose of the tool on the end-effector using a kinematic model of the robot arm.

Movement of the tool can be adjusted with high accuracy by the feedback control loop. The feedback control loop rejects the input disturbance, d_{q_m} , and minimizes the

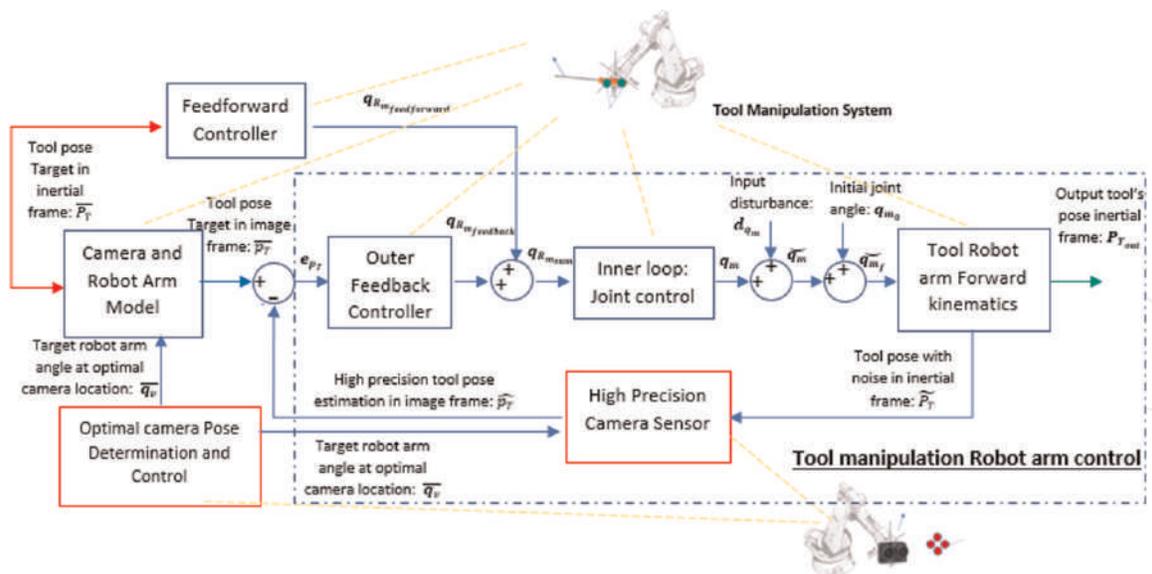


Figure 8.
 The high-accuracy tool manipulation control block diagram.

error between the tool pose target, \overline{p}_T , in the image frame and the high precision estimation tool pose from the camera sensor, \widehat{p}_T . The pose in a robot system modeled in Cartesian inertial base frame consists of six degrees of freedom, i.e., three translations and three rotations. Therefore, in order to have a full control of the tool pose, the camera in the feedback control loop requires to measure the image coordinates of at least two interests points on the tool.

The feedforward and feedback controllers work simultaneously to move the tool to the target pose location in the tool manipulation system. The combined target $q_{R_{sum}}$ are the inputs to the joint control loop so that both controllers manipulate the tool pose. The benefit of designing both feedback and feedforward controls for the manipulation system is to reduce the time duration. If only feedback control is utilized, the pose estimation generated from the visual system requires taking multiple pictures and makes the tool movement very slow. We can divide the task of the tool movement control into two stages. In the first stage, under the action of the feedforward control, the tool moves to an approximate location that is close to the desired destination. In the second stage, the feedback controller moves the tool to the precise target location using the tool pose estimation from the camera. In addition, the camera has a range of view and can only detect the tool and measure its 2D feature as \widehat{p}_T when it is not far away from the target. When the tool is moving from a location that is not in the camera range of view, we must estimate the feature as \widetilde{p}_T until the tool moves into the range of view (this point will be discussed in detail in Section 7). It should be noted that only the feedback controller has the ability to compensate for uncertainties.

This control topology is an analogy to the macro-micro manipulation in the current industry trends where the large-scale robots are used for the approximate positioning, while the small-scale robots are utilized for the precise positioning [2].

4.5 The high precision camera sensor model

As shown in **Figures 7 and 8**, the high precision camera sensor model provides high precision estimations in the feedback loop of the tool manipulation system

control and the camera movement adjustment control. The camera robot arm model, which is shown in both **Figures 7** and **8**, is the target generator that transforms the target in the inertial frame to the target locations in the image frame. The mathematical model of the camera, which is utilized in the visual robot arm and in the feedback loop to generate the required position estimation, has an equivalent Hardware-In-the-Loop (HIL) model as shown in **Figure 9**.

Figure 9 shows the details of high precision camera model and its equivalent HIL model. The upper configuration is the mathematical model that is used in the simulation to generate image coordinates and to design the outer-loop controller in the robot arm control loop. However, in real application, the lower HIL configuration replaces this mathematical model. In the HIL model, the image processing will make an estimation of the tool pose with high precision.

5. The inner-loop control design

In this part, we will present the control system design of the camera pose control and the inner joint control loops for both the camera movement adjustment and the tool manipulation, which have been introduced in Section 4. A simulation scenario is also presented in this section.

We express an equation for the 6 DOF manipulator including the dynamics of the robot manipulator and the actuators (DC motor) in the following simplified form [28]:

$$(D(q) + J)\ddot{q} + \left(C(q, \dot{q}) + \frac{B}{r} \right) \dot{q} + g(q) = u \quad (19)$$

where $D(q)$ and $C(q, \dot{q})$ are 6×6 inertial and Coriolis matrices respectively. J is a diagonal matrix expressing the sum of actuator and gear inertias. B is the damping factor, and r is the gear ratio, $g(q)$ is the term for potential energy, u is the 6×1 input

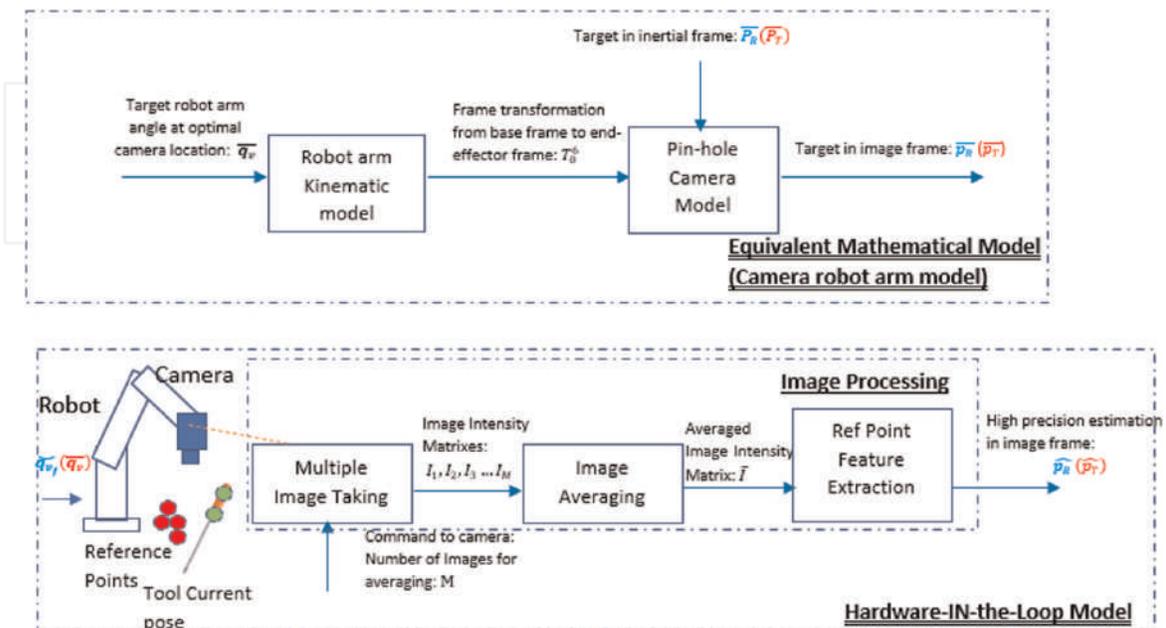


Figure 9.

The high precision camera sensor. Note: Blue are signals used in the control system of **Figure 7** and red are signals used in the control system of **Figure 8**.

vector, and q is the 6×1 generalized coordinates (in this paper, q is the 6×1 joint angle vector).

We can simplify Eq. (19) as follows:

$$M(q)\ddot{q} + h(q, \dot{q}) = u \quad (20)$$

with

$$M(q) = D(q) + J \quad (21)$$

$$h(q, \dot{q}) = \left(C(q, \dot{q}) + \frac{B}{r} \right) \dot{q} + g(q) \quad (22)$$

If we transform the control input as following:

$$u = M(q)v + h(q, \dot{q}) \quad (23)$$

where v is a virtual input. Then we substitute for u in Eq. (20) using (23), and since $M(q)$ is invertible, we will have a reduced system equation as follows:

$$\ddot{q} = v \quad (24)$$

This transformation is so-called feedback linearization technique with the new system equation given in Eq. (24). This equation represents 6 uncoupled double integrators. The overall feedback linearization method is illustrated in **Figure 10**. In this control block diagram, we force the joint angle q to follow the target joint angle q_R so that the output pose, P , can follow the target pose, \bar{P} . P , \bar{P} , q , and q_R are all vectors with six elements (each element corresponds to a joint position or angle). The Nonlinear interface transform the linear virtual control input v to the nonlinear control input u by using Eq. (23). The output of the manipulator dynamic model, the joint angles, q , and their first derivatives, \dot{q} , are utilized to calculate $M(q)$ and $h(q, \dot{q})$ in the Nonlinear interface. The linear joint controller is designed using Youla parameterization technique [9] to control the nominally linear system in Eq. (24).

The design of a linear Youla controller with nominally linear plant is presented next.

Since the transfer functions between all inputs to outputs in (24) are the same and decoupled, we can first design a SISO (single input and single output) controller and use the multiple of the same controller for a six-dimension to obtain the MIMO (Multiple Input Multiple Output) version. In other words, we first design a controller G_{cSISO} that satisfies:

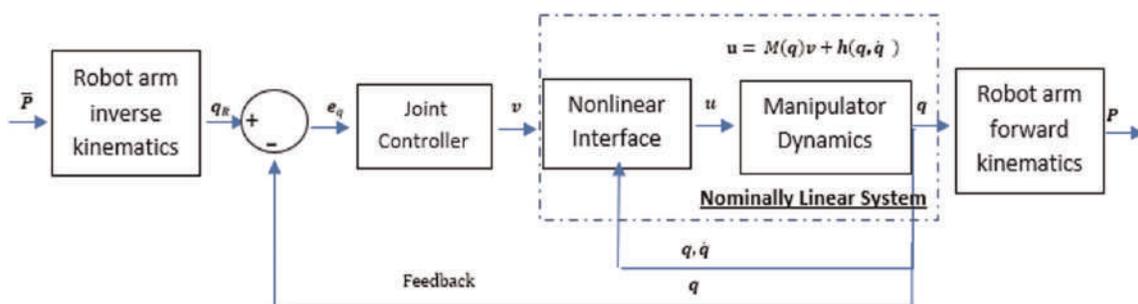


Figure 10.
 The block diagram of feedback linearization Youla control design used for the joint control loop.

$$v_{SISO} = \ddot{q}_{SISO} \quad (25)$$

where v_{SISO} is a single input to a nominally linear system and \ddot{q}_{SISO} is the second order derivative of a joint angle. The controller system in **Figure 10** can be then written as:

$$G_{c_{sys}} = G_{c_{SISO}} \cdot I \quad (26)$$

where I is a 6×6 identity matrix. We can design the SISO controller G_c using Youla parameterization technique [48]. The transfer function of the SISO nominally linear system from (24) is:

$$G_{p_{SISO}} = \frac{1}{s^2} \quad (27)$$

Note that $G_{p_{SISO}}$ has two BIBO (Bounded Input Bounded Output) unstable poles at origin. To ensure internal stability of the feedback loop, the closed loop transfer function, T_{SISO} , should meet the interpolation conditions [39]:

$$T_{SISO}(s = 0) = 1 \quad (28)$$

$$\left. \frac{dT_{SISO}}{ds} \right|_{s=0} = 0 \quad (29)$$

We compute a Youla transfer function: Y_{SISO} , using the following relationship,

$$T_{SISO} = Y_{SISO} G_{p_{SISO}} \quad (30)$$

The T_{SISO} is designed so that it satisfies the conditions in (28) and (29). The sensitivity transfer function, S_{SISO} , is then calculated as follows:

$$S_{SISO} = 1 - T_{SISO} \quad (31)$$

Without providing the design details, we refer the interested reader to [38], the closed-loop transfer function should be in the following form to satisfy the interpolation conditions:

$$T_{SISO} = \frac{(3\tau s + 1)}{(\tau s + 1)^3} \quad (32)$$

Where τ specifies the pole and zero locations and represents the bandwidth of the control system. We can tune τ so that the response can be fast with less-overshoot.

Then we can derive $G_{c_{SISO}}$ from relationships between the closed-loop transfer function, T_{SISO} , the sensitivity transfer function, S_{SISO} , and the Youla transfer function, Y_{SISO} , in Eqs. (30)–(32):

$$Y_{SISO} = T_{SISO} G_{p_{SISO}}^{-1} = \frac{s^2(3\tau^2 s + 1)}{(\tau s + 1)^3} \quad (33)$$

$$S_{SISO} = 1 - T_{SISO} = \frac{s^2(\tau^3 s + 3\tau^2)}{(\tau s + 1)^3} \quad (34)$$

$$G_{cSISO} = Y_{SISO} S_{SISO}^{-1} = \frac{3\tau^2 s + 1}{\tau^3 s + 3\tau^2} \quad (35)$$

From Eq. (35), we can compute a MIMO controller as follows:

$$G_{c_{sys}} = \frac{3\tau^2 s + 1}{\tau^3 s + 3\tau^2} \cdot I_{6 \times 6} \quad (36)$$

Eq. (36) provides the desired controller, which is used as the joint controller, as shown in **Figure 10**. This configuration is precisely the inner joint control loop in both the visual and the manipulator systems as shown in **Figures 7 and 8**.

Figure 11 shows the simulation results for the case with no disturbance. The target position and the orientation of the end-effector are selected to be $\begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} = \begin{bmatrix} 1.7157m \\ 1.0191m \\ 0.7518m \end{bmatrix}$

and $\begin{bmatrix} \bar{n} \\ \bar{s} \\ \bar{a} \end{bmatrix} = \begin{bmatrix} -0.425 & 0.87 & 0.25 \\ 0.8361 & 0.2714 & 0.4767 \\ 0.3469 & 0.4116 & -0.8428 \end{bmatrix}$, where $[\bar{X}, \bar{Y}, \bar{Z}]^T$ is the absolute position

coordinate of the center of the end effector in the inertial frame and $\bar{n}, \bar{s}, \bar{a}$ represent respectively the end-effector's directional unit vector of the yaw, pitch and roll in the inertial frame. Therefore, the corresponding target angles of rotations are $q_R = [30^\circ, 60^\circ, -45^\circ, 15^\circ, 45^\circ, 90^\circ]$. For this simulation, we have designed the control system with the bandwidth of 100 rad/s . In the following three plots, solid lines represent the responses for the end-effector position of each joint and the end-effector orientation respectively, and the dashed lines are the targets. Specifically, the orientation response of the end-effector is the vector that tangent to the curve in the second plot at each point in **Figure 11**. The simulation results show that all responses of the controlled system will be able to reach their final/steady state values within 0.1 second with no (or little) overshoots.

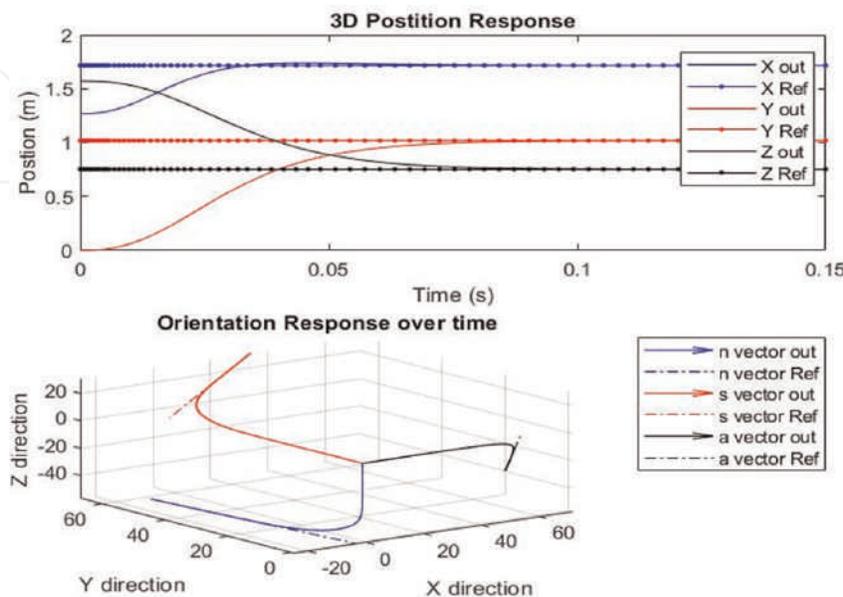


Figure 11.
 The simulation results for the end-effector response to an arbitrary selected trajectory.

6. The cascaded SISO outer-loop design for the camera movement adjustment control system

As introduced in Section 4.1, we use the Image Based Visual Stereo (IBVS) as the framework for cascaded control design of both the visual and the manipulation system. The inner-loop is the joint control loop, as discussed previously. In the IBVS, an outer feedback control is designed in addition to the inner feedback control so that the 2D visual features can be compared and matched. Therefore, a camera model is required in the feedback loop to map the joint angles to the object visual features (e.g., 2D coordinates of the object in the image frame). To start from a simple case, we discuss the requirements for the design of the cascaded control for a SISO system. Assume, we only measure and try to control one feature: the coordinate of the object in one axis, then the overall cascaded control diagram of this configuration is shown in **Figure 12**.

\bar{u}_R is the target of u coordinate of a point (one reference point in visual system). $G_{c_{out}}$ is the outer-loop controller, which provides the target joint angle q_{R_v} to the robot arm based on the image coordinate error e_{u_R} . A combined uncertainty signal, d_{q_v} (e.g., sensor noise, backlash, friction, and compliance due to gear reduction in the joint) is added to the joint angle of rotation q_v , the output of the inner control loop. Furthermore, the initial joint angle of robot manipulator, q_{v_0} , is added to the output joint angle with disturbance, \bar{q}_v , to generate the final joint angle \bar{q}_{v_f} .

6.1 The camera sensor model for the SISO control design

In this section, we are going to present a simplified mathematical model of the camera. Let us consider a one-link robot manipulator with a camera that can be rotated around the Z -axis, as shown in **Figure 13**. Let us also consider a point that is located in the $X - Y$ plane.

We use a Cartesian coordinate frame $OXYZ$, which is attached at the camera center. The X -axis is perpendicular to the camera lens and the coordinate frame rotates around Z -axis. A point whose coordinate is $(X_P, Y_P, 0)$ in the rotational frame is projected on the image $u - v$ plane with its coordinates as $(u, 0)$. In addition, an inertial frame $\bar{O}\bar{X}\bar{Y}\bar{Z}$ is shown at the base of the manipulator in **Figure 13**.

We derive the equations for a pin-hole camera model (a monocular case in stereo model in Section 2.1). Assuming the skew coefficient and image coordinate offsets are zero ($s_c = u_0 = v_0 = 0$), then from Eq. (1), the one-dimensional image coordinate u can be written as:

$$u = \frac{f_u Y_P}{X_P} \quad (37)$$

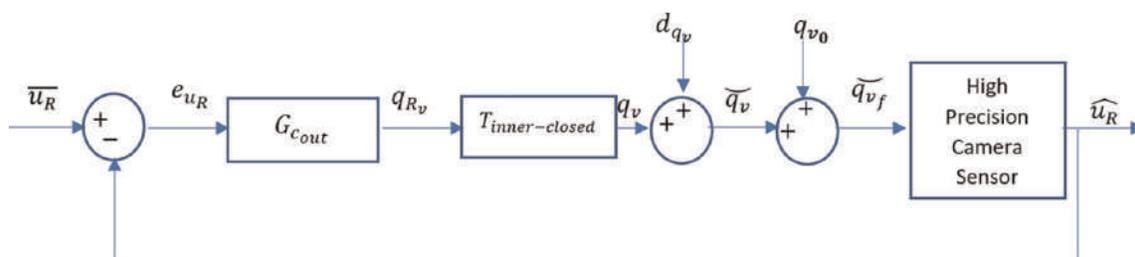


Figure 12. Control block diagram for the SISO camera movement adjustment system.

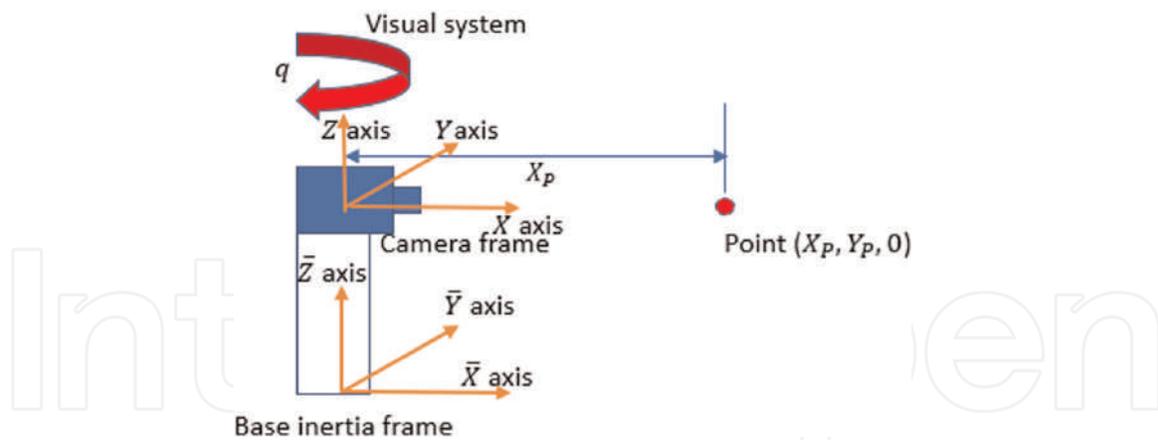


Figure 13.
 One-link manipulator with a camera.

The coordinate system, located at the camera center, rotates with the camera. Assume the camera starts at the position where the X -axis is parallel to the \bar{X} -axis of the inertial coordinate system and positive angle is defined when the camera rotates in a clockwise direction with an angle q around Z -axis. After the frame rotates q clockwise, the new coordinates of the point in the new coordinate frame $(X'_p, Y'_p, 0)$ can be computed as:

$$\begin{bmatrix} X'_p \\ Y'_p \end{bmatrix} = \begin{bmatrix} \cos(q) & -\sin(q) \\ \sin(q) & \cos(q) \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \end{bmatrix} \quad (38)$$

Combining Eqs. (37) and (38), the new image coordinate, u' , can be calculated as:

$$u' = \frac{f_u Y'_p}{X'_p} = f_u \frac{X_p \sin(q) + Y_p \cos(q)}{X_p \cos(q) - Y_p \sin(q)} \quad (39)$$

Let us write the coordinates (X_p, Y_p) in the polar coordinates, such that,

$$R = \sqrt{X_p^2 + Y_p^2}$$

and

$$X_p = R \cos(\varphi), Y_p = R \sin(\varphi),$$

where

$$\varphi = \tan^{-1}(Y_p/X_p) \quad (40)$$

Therefore, Eq. (39) can be written in the polar form as:

$$u' = f_u \frac{R \sin(\varphi + q)}{R \cos(\varphi + q)} = f_u \tan(\varphi + q) \quad (41)$$

The angle φ is the angle of the point with respect to the \bar{X} -axis and the angle q is already defined. Eq. (41) mathematically expresses a model of the camera sensor shown in the block diagram of **Figure 13**.

6.2 Outer-loop controller design

In the previous sections, we presented the results for designing inner joint loop controllers using Youla parameterization method. In the following two sections, we will discuss the design of the outer-loop controllers by using two different approaches:

- a. Feedback linearization
- b. Model linearization

6.2.1 Feedback linearization

Let us relook at the cascaded block diagram in **Figure 12**. The inner closed loop transfer function is already derived in (32):

$$T_{inner-closed} = \frac{q_v(s)}{q_{R_v}(s)} = \frac{(3\tau_{in}s + 1)}{(\tau_{in}s + 1)^3} \quad (42)$$

Notice we replaced τ with τ_{in} to indicate bandwidth of inner-closed loop. Using Eq. (41) and considering the block diagram in **Figure 12**, we can write:

$$\widehat{u}_R = f_u \tan(\varphi + q_{vf}), \text{ with } \varphi = \tan^{-1}(Y_P/X_P) \quad (43)$$

We can rewrite (42) and (43) in time domain, by introducing an intermediate variable or state, W , as:

$$\tau_{in}^3 \dddot{W} + 3\tau_{in}^2 \ddot{W} + 3\tau_{in} \dot{W} + W = q_{R_v} \quad (44)$$

$$f_u \tan(\varphi + q_{v0} + 3\tau_{in} \dot{W} + W) = \widehat{u}_R \quad (45)$$

Eqs. (44) and (45) describe a nonlinear third order system, where W is the state, q_{R_v} is the input and \widehat{u}_R is the output. We can use feedback linearization method to design the outer-loop controller by taking the second order derivative of (45) and combine with (44) to obtain:

$$\ddot{\widehat{u}}_R = R(W, \dot{W}, \ddot{W}) + G(W, \dot{W}, \ddot{W})q_{R_v} \quad (46)$$

where

$$R(W, \dot{W}, \ddot{W}) = 2f_u \cos^{-2}(\varphi + q_{v0} + 3\tau_{in} \dot{W} + W) \tan(\varphi + q_{v0} + 3\tau_{in} \dot{W} + W) \\ (3\tau_{in} \ddot{W} + \dot{W})^2 - f_u \cos^{-2}(\varphi + q_{v0} + 3\tau_{in} \dot{W} + W) \left(8\ddot{W} + \frac{9}{\tau_{in}} \dot{W} + \frac{3}{\tau_{in}^2} W \right) \quad (47)$$

$$G(W, \dot{W}, \ddot{W}) = f_u \cos^{-2}(\varphi + q_{v0} + 3\tau_{in} \dot{W} + W) \frac{3}{\tau_{in}^2} \quad (48)$$

We can transform or map these variables so that the nonlinear system in (47)–(48) can be written as an equivalent linear state-space representation as follows:

$$\varepsilon_1 = f_u \tan(\varphi + q_{v0} + 3\tau_{in}\dot{W} + W) \quad (49)$$

$$\varepsilon_2 = \dot{\varepsilon}_1 = f_u \cos^{-2}(\varphi + q_{v0} + 3\tau_{in}\dot{W} + W) \cdot (3\tau_{in}\ddot{W} + \dot{W}) \quad (50)$$

We can write the state-space form of (49) and (50) as:

$$\dot{\varepsilon} = \begin{bmatrix} \dot{\varepsilon}_1 \\ \dot{\varepsilon}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U \quad (51)$$

$$\widehat{u}_R = [1 \ 0] \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \quad (52)$$

where

$$U = G(W, \dot{W}, \ddot{W})q_{R_v} + R(W, \dot{W}, \ddot{W}) \quad (53)$$

Transform the state-space representation back to the transfer function form, we can write:

$$Gp_{nominal} = \frac{\widehat{u}_R(s)}{U(s)} = C(sI - A)^{-1}B = \frac{1}{s^2} \quad (54)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } C = [1 \ 0] \quad (55)$$

Since the $Gp_{nominal}$ is the same as the plant transfer function in (27), the design of Youla controller for this linear system is similar to (28)–(35). Therefore, the transfer function of the outer-loop controller can be written as:

$$G_{c_{out}} = \frac{3\tau_{out}^2s + 1}{\tau_{out}^3s + 3\tau_{out}^2} \quad (56)$$

τ_{out} determines the pole and zero locations of closed-loop transfer function of the outer-loop controller and therefore, represents the bandwidth of outer-loop controller. We must make sure that $\tau_{out} > \tau_{in}$ so that inner-loop responds faster than the outer-loop in the cascaded control design strategy. The overall block diagram is shown in **Figure 14**.

The states W, \dot{W}, \ddot{W} are computed from the rotational angle and its derivative \widetilde{q}_{v_f} and $\dot{\widetilde{q}}_{v_f}$. From Eq. (42), we can obtain the following relationship:

$$\widetilde{q}_{v_f} = 3\tau_{in}\dot{W} + W \quad (57)$$

Therefore, the transfer function from W to \widetilde{q}_{v_f} can be written as:

$$W(s) = \frac{1}{3\tau_{in}s + 1} \widetilde{q}_{v_f}(s) \quad (58)$$

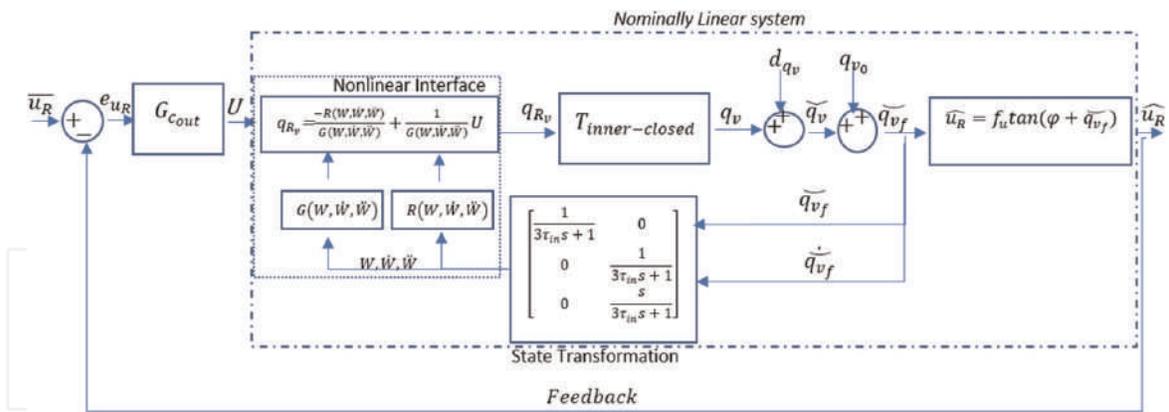


Figure 14. The block diagram of the SISO outer-loop control with feedback linearization for robot arm movement adjustment.

Also, we can obtain the transfer function of $\dot{W}(s)$ and $\ddot{W}(s)$ as:

$$\dot{W}(s) = \frac{1}{3\tau_{in}s + 1} \dot{q}_{v_f}(s) \quad (59)$$

$$\ddot{W}(s) = \frac{s}{3\tau_{in}s + 1} \dot{q}_{v_f}(s) \quad (60)$$

Eqs. (58)–(60) are shown as the state transformation block in **Figure 14**. It is worthwhile to note that the equation in the nonlinear interface:

$$q_{R_v} = \frac{-R(W, \dot{W}, \ddot{W})}{G(W, \dot{W}, \ddot{W})} + \frac{1}{G(W, \dot{W}, \ddot{W})} U \quad (61)$$

is only defined when $G(W, \dot{W}, \ddot{W}) = f_u \cos^{-2}(\varphi + q_{v_0} + q_v) \frac{3}{\tau_{in}^2} \neq 0$. It can be shown that this is always not equal to zero and $\varphi + q_{v_0} + q_v \neq \pm \frac{\pi}{2}$. This is always true because any angle of rotation should be within half of camera's angle of view α ; that is $|\varphi + \widetilde{q}_{v_f}| \leq \frac{\alpha}{2} < \frac{\pi}{2}$. In addition, α is always less than π for any camera type. Therefore, the controller works for the entire range independent of the camera type.

6.3 Model linearization

We can deal with the nonlinear system by linearizing the system first and then design a linear controller using the system transfer function. In **Figure 12**, $T_{inner-closed}$ transfer function is given in (42) and the nonlinear form of the camera model is provided in (43). The overall dynamic system combines the inner-loop and the camera model, which will be linearized so that the combined dynamic system will then be linear. Next, we linearize, the camera model, (43), around an equilibrium point \widetilde{q}_{v_0} :

$$\widehat{u}_R = f_u \cos^{-2}(\varphi + \widetilde{q}_{v_0}) (\widetilde{q}_{v_f} - \widetilde{q}_{v_0}) + f_u \tan(\varphi + \widetilde{q}_{v_0}) \quad (62)$$

If we assume $\widetilde{q}_{v_0} = 0$, then:

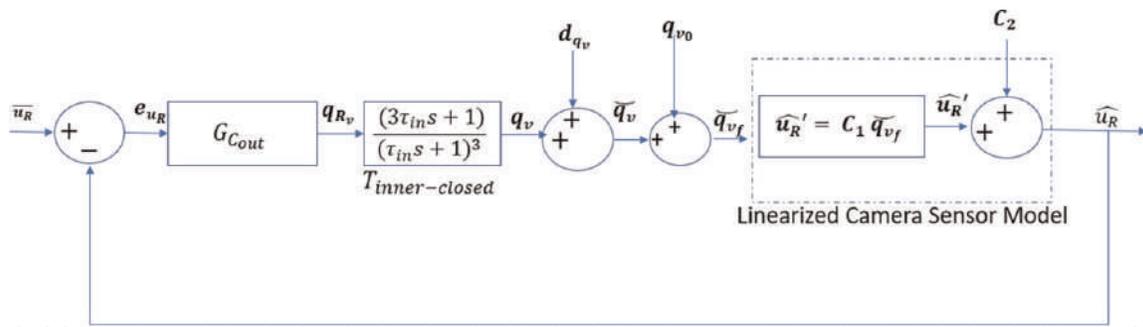


Figure 15. The block diagram of the SISO control with the linearized camera model for robot arm movement adjustment.

$$\widehat{u}_R = f_u \cos^{-2}(\varphi) \widetilde{q}_{v_f} + f_u \tan(\varphi) \quad (63)$$

Assuming $C_1 = f_u \cos^{-2}(\varphi)$, $C_2 = f_u \tan(\varphi)$, therefore, Eq. (63) can be rewritten as:

$$\widehat{u}_R = C_1 \widetilde{q}_{v_f} + C_2 \quad (64)$$

Let us define $\widehat{u}_R' = \widehat{u}_R - C_2$, then, the overall block diagram of the linearized system is shown in **Figure 15**.

The plant transfer function is derived as:

$$G_{p_{linear}} = \frac{\widehat{u}_R'}{q_{R_v}} = C_1 \frac{(3\tau_{in}s + 1)}{(\tau_{in}s + 1)^3} \quad (65)$$

The design of a Youla controller is trivial in this case as all poles/zeros of the plant transfer function in (65) are located in the left half-plane, and therefore, they are stable. In this case, we can shape the closed loop transfer function, T_{out} , by selecting a Youla transfer function: Y_{out} so that the closed loop transfer function, T_{out} , will not contain any plant poles and zeros. All poles and zeros in the original plant can be canceled out and new poles and zeros can be added to shape the closed-loop system. Let us select a Youla transfer function so that the closed-loop system behaves like a second order Butterworth filter, such that:

$$Y_{out} = \frac{1}{G_{p_{linear}}} \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (66)$$

then:

$$T_{out} = \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (67)$$

where ω_n is called natural frequency and approximately sets the bandwidth of the closed-loop system. We need to make sure the bandwidth of the outer-loop is smaller than the inner-loop, i.e., $1/\omega_n > \tau_{in}$. ζ is called the damping ratio, which is another tuning parameter.

Using Eqs. (34) and (35), we can calculate the sensitivity transfer function, S_{out} , and the controller transfer function, $G_{C_{out}}$, of the outer-loop in cascaded control design as:

$$S_{out} = 1 - T_{out} = \frac{s^2 + 2\zeta\omega_n s}{(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (68)$$

$$G_{c_{out}} = Y_{out} S_{out}^{-1} = \frac{1}{C_1} \frac{(\tau_{in} s + 1)^3}{(3\tau_{in} s + 1)} \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s)} \quad (69)$$

6.4 The simulation of the cascaded SISO closed-loop system for camera movement adjustment

In this section, we are going to compare the closed-loop response results of the cascaded control system where the outer-loop controllers are designed using the two aforementioned methods: feedback linearization (Section 6.2) and model linearization (Section 6.3). The simulation results are obtained with the original nonlinear camera model (43). In addition, for the linearized plant approach, we will also illustrate how varying the damping ratio ζ affects the responses. For both methods, we chose the bandwidth of the inner-loop as $100rad/s$ and the bandwidth of the outer-loop as $10rad/s$.

We compare the simulation responses by choosing six different damping ratios. Two are chosen as the overdamped systems ($\zeta > 1$), one is chosen as a critically damped system ($\zeta = 1$), and three as the underdamped systems ($\zeta < 1$). We have simulated four cases and compared all six systems for each case. Each case is different due to varying the initial angle φ (see Eq. (40)) and the input disturbance d_{q_v} . Two cases are simulated without the input disturbance while the other two are simulated with the disturbance to compare the robustness of the controlled system.

The step responses of the image coordinate \hat{u}_R is shown in **Figures 16–19**. The intrinsic camera parameters are selected to be: $f_u = 2.8 mm$ (Focal length) and $\alpha = 120^\circ$ (Angle of view). In cases 1 and 2, the responses of feedback linearization are displayed in black dashed lines while all other lines are the responses of the controller that is designed with the linearized plant and varying the damping ratio ζ .

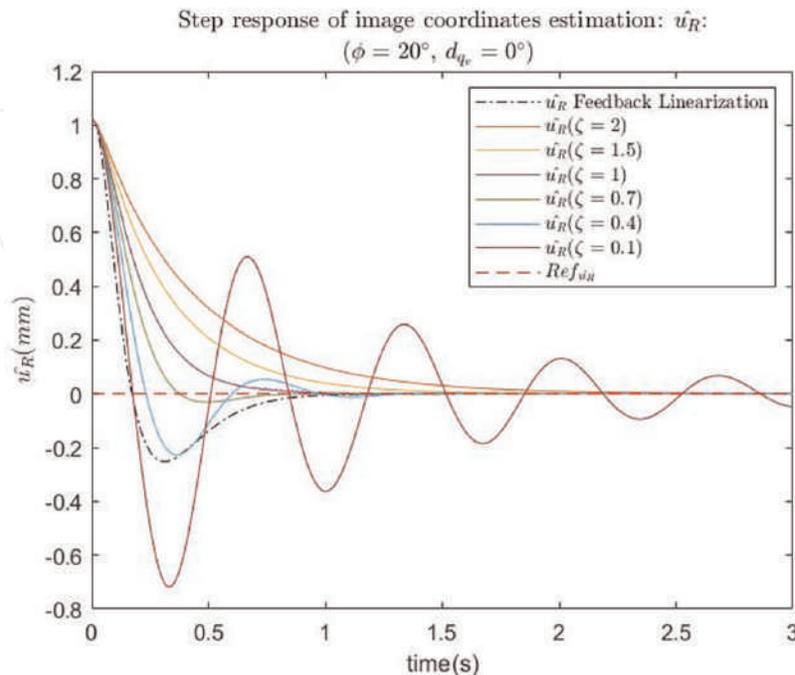


Figure 16. Step responses of \hat{u}_R for the case 1: $\varphi < \frac{\alpha}{2} = 20^\circ, d_{q_v} = 0^\circ$.

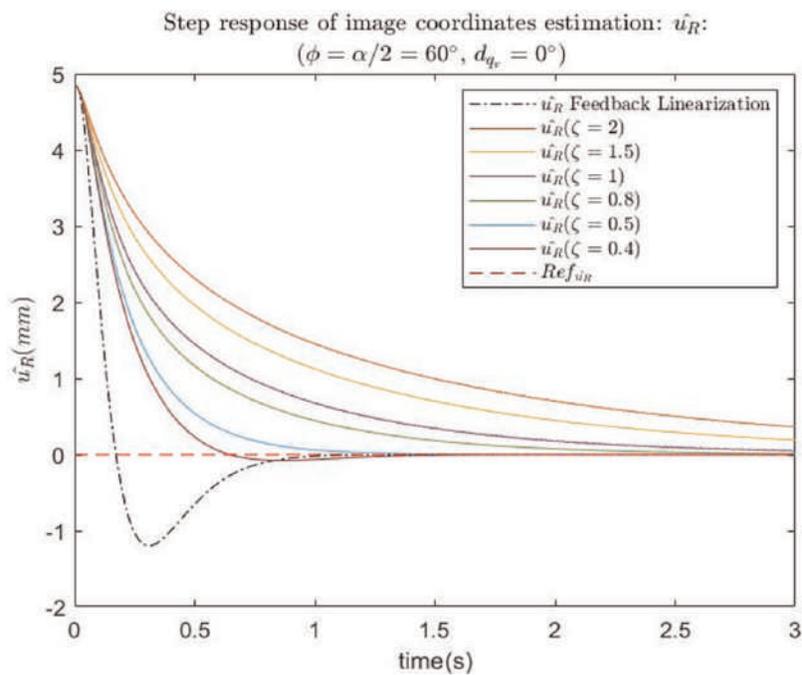


Figure 17.
 Step responses of \hat{u}_R for the case 2: $\varphi = \frac{\alpha}{2} = 60^\circ, d_{qv} = 0^\circ$.

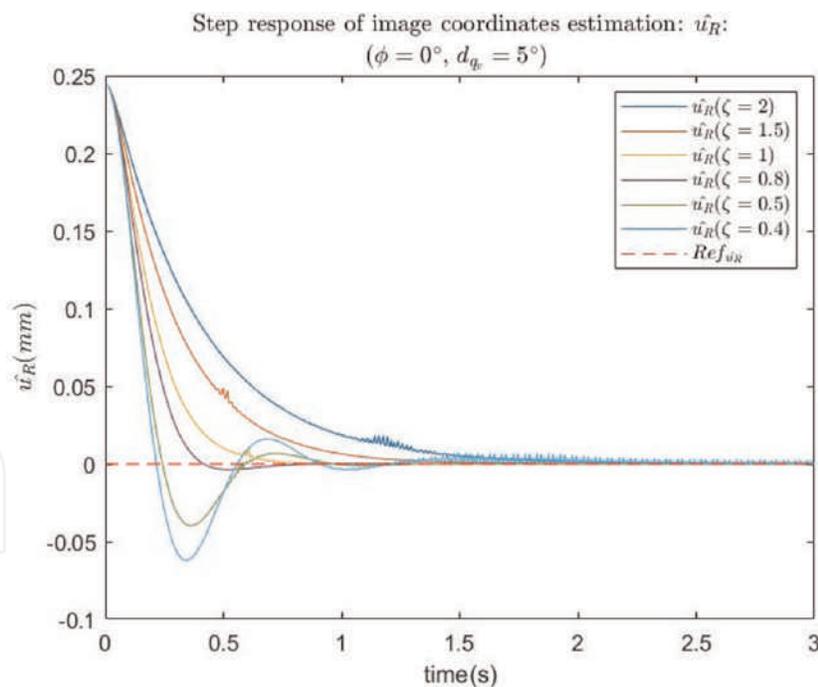


Figure 18.
 Step responses of \hat{u}_R for the case 3: $\varphi = 0^\circ, d_{qv} = 5^\circ$.

The case 1 is simulated with the initial angle $\varphi < \frac{\alpha}{2}$, while the case 2 is simulated when $\varphi = \frac{\alpha}{2}$, the largest possible initial angle within the angle of view. It can be shown clearly that without any input disturbance, both methods are able to drive the closed-loop responses to the final value. The step response of the feedback linearization has an overshoot. In addition, for the second method, the model linearization approach, it can be seen from the two simulation cases that there exists a damping ratio, ζ_{opt} , such that.

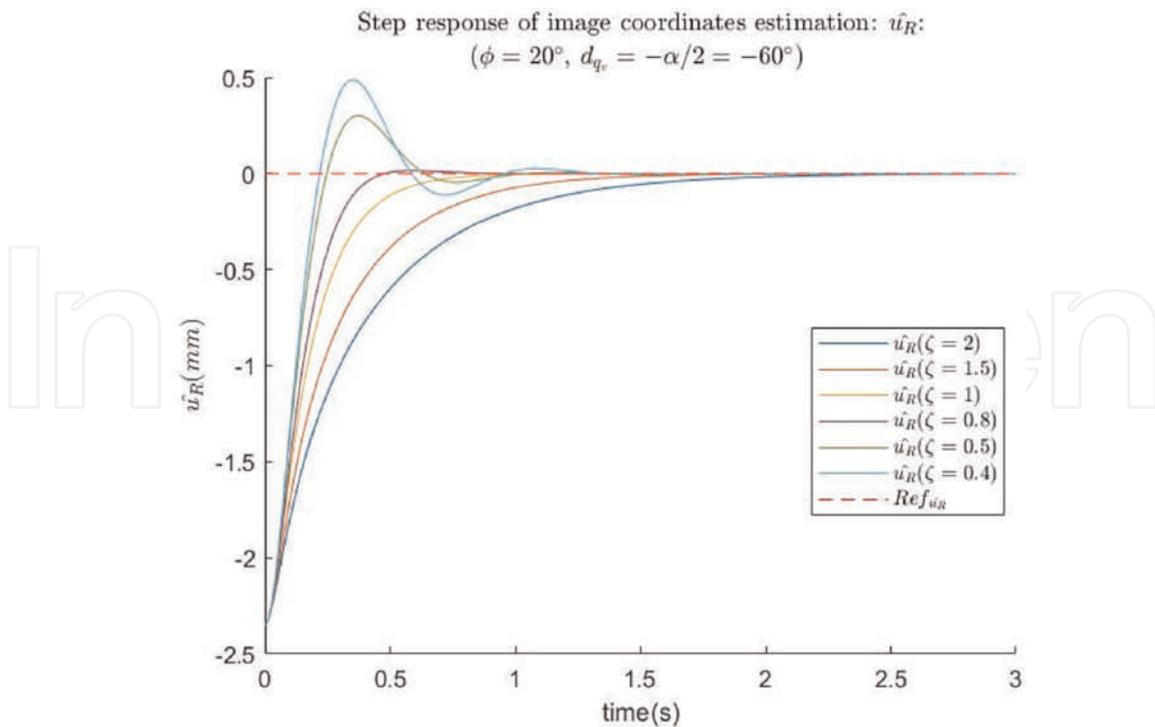


Figure 19. Step responses of \hat{u}_R for the case 4: $\varphi = 20^\circ, d_{q_v} = -\frac{\alpha}{2} = -60^\circ$.

- When $\zeta \geq \zeta_{opt}$, the step responses have no overshoots and as ζ decreases, the system reaches the steady state faster.
- When $\zeta < \zeta_{opt}$, the step responses have overshoots, and the overshoots increase as ζ decreases. As ζ increases, the system reaches the steady state faster.

It can be estimated from **Figures 16** and **17** that $\zeta_{opt} \cong 1$ in the case 1 and $\zeta_{opt} \cong 0.5$ in the case 2. The most desirable system is the one without overshoot and fastest step response. When $\zeta = \zeta_{opt}$, the system has the fastest response and no (or little) overshoot. Therefore, we can state that the best performance of the controlled system is when setting the damping ratio $\zeta = \zeta_{opt}$. Clearly, the value of ζ_{opt} varies with φ , the initial angle of the reference point with respect to the inertial frame.

In the cases 3 and 4, the input disturbance is introduced to the system. In the case 3, a small disturbance ($d_{q_v} = 5^\circ$) is added to the actuator input. The case 4 is a combined case where both φ and d_{q_v} are present ($\varphi = 20^\circ, d_{q_v} = -60^\circ$). **Figures 18** and **19** do not display the step responses of feedback linearization approach. The step responses of feedback linearization are unstable when input disturbances are introduced. It can be shown that any input disturbance drastically alters the nonlinear interface parameters, used in feedback linearization, and hence, results in an unstable system. On the other hand, the linear controller designed based on the linearized plant model is robust to the input disturbances even with the significantly large disturbances (case 4). Similar to the no disturbance cases, ζ_{opt} exists for the cases with disturbances.

From the discussion above, the plant linearization method is the preferred and the recommended method for the camera movement adjustment.

7. The cascaded SISO outer-loop design for high-accuracy robot tool manipulator

The control block diagram of high-accuracy robot tool manipulator is shown in **Figure 8**. In this diagram, the camera is kept static but serves the movement of the tool by the visual data. This block diagram contains a feedback loop as well as a feedforward loop. We design controllers for each loop and simulate the combined loop under different scenarios.

7.1 Developing a combined SISO tool robot arm and camera model

In **Figure 8**, the joint angle including the disturbance of the tool manipulator is transformed to the tool pose using a robot kinematics model. A camera model then is utilized to convert the 3D pose to the 2D, as shown in **Figure 20**. For simplicity, we

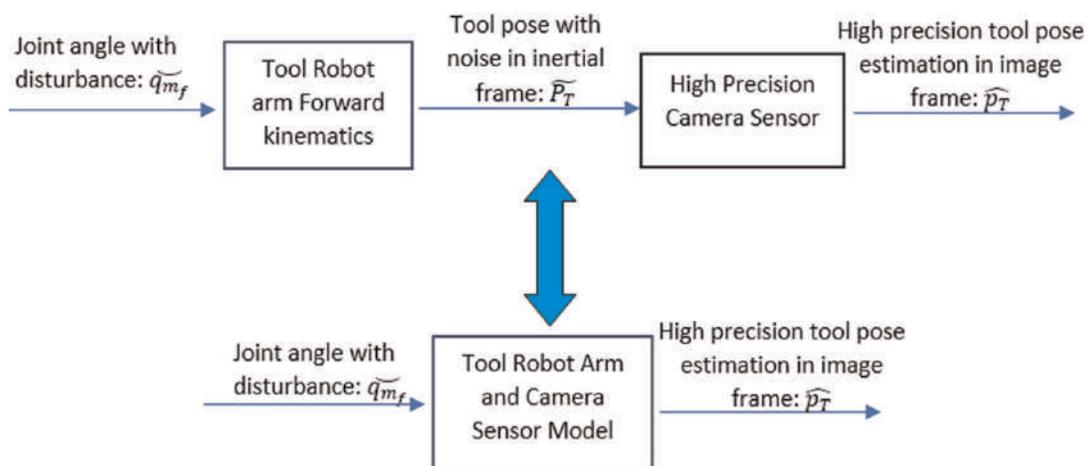


Figure 20.
 The tool robot arm and the camera sensor model (a combined block of tool manipulation kinematics and camera sensing).

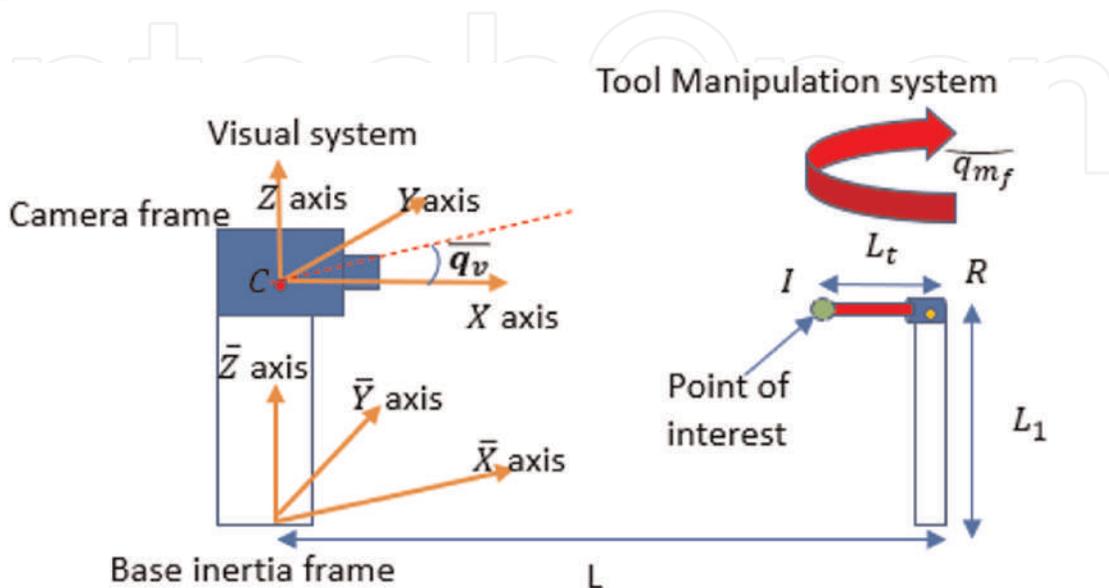


Figure 21.
 A SISO camera and tool robot arm setup.

can combine these two blocks into one block, which is called the tool robot arm and camera sensor model.

In **Figure 21**, a SISO combined model setup has been shown based on the one-link camera robot arm model in **Figure 13**. Now, the camera, which is attached to a one-link rotational robot arm captures the image of a tool, which is attached to another similar robot arm, and estimates its angle of rotation. The tool has a length L_t with an interest point is selected at the tip of the tool. Both robot links have a length of L_1 and are separated from each other by a distance L . Assume \bar{q}_v is the angle of the camera from previous control sequences (discussed in **Figure 5**). The inertial and camera coordinate frames set ups are discussed in Section 6.1. The only difference is that the camera frame rotates relative to inertial frame by a clockwise angle \bar{q}_v along the \bar{Z} -axis. The tool rotates relative to the \bar{Z} -axis in a clockwise direction with a variable angle \bar{q}_m . The actual angle of rotation \bar{q}_m is the sum of the input disturbance d_{q_m} and the planned angle of rotation q_m ; i.e.,

$$\bar{q}_m = q_m + d_{q_m} \quad (70)$$

Then, we can compute the final angle after rotation by adding the initial angle of the tool in the inertial frame q_{m_0} :

$$\bar{q}_{m_f} = \bar{q}_m + q_{m_0} \quad (71)$$

The coordinates of the point of interest on the tool in the inertial frame is then computed as $(L - L_t \cos(\bar{q}_{m_f}), L_t \sin(\bar{q}_{m_f}), L_1)$

Following the same procedures as in Eqs. (37)–(41), we can derive the tool image coordinate \hat{u}_T along the u -axis as:

$$\hat{u}_T = f_u \frac{Q(\bar{q}_{m_f}) + \tan(\bar{q}_v)}{1 - Q(\bar{q}_{m_f}) \tan(\bar{q}_v)} \quad (72)$$

where,

$$Q(\bar{q}_{m_f}) = \frac{L_t \sin(\bar{q}_{m_f})}{L - L_t \cos(\bar{q}_{m_f})} \quad (73)$$

Eqs. (72) and (73) provide a function that maps the current or the final angle of the tool onto the image coordinate \hat{u}_T with constant parameters, \bar{q}_v , L , and L_t .

7.2 The outer-loop feedback and feedforward controller design

The overall plant for the design of this control system is composed of the inner joint control loop, see Eq. (32) and **Figure 8**, and the tool robot arm and camera sensor models, as shown in **Figure 20**. We can design the outer-loop feedback controller using the feedback linearization method or the plant linearization method by following the procedures presented in Section 6.2 and 6.3 respectively. For the sake of brevity, we will not discuss the detail derivations of each controller. Mostly

comparative issues such the overshoots and the robustness are discussed in Section 6.4. In this section, we utilize the plant linearization method to design the outer-loop feedback controller.

Without providing the details, the controller is designed for a second order closed-loop system using the plant linearization method (the plant is linearized at $\bar{q}_{mf} = 0^\circ$) is given as:

$$G_{c_{out}} = \frac{1}{C_1} \frac{(\tau_{in}s + 1)^3}{(3\tau_{in}s + 1)} \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s} \quad (74)$$

and

$$C_1 = f_u (1 + (\tan(\bar{q}_v))^2) \frac{LL_t - L_t^2}{(L - L_t)^2} \quad (75)$$

Then, the second order closed-loop transfer function T of the overall cascaded control system is expressed as:

$$T = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (76)$$

Where f_u is the camera focal length, \bar{q}_v, L , and L_t are the parameters defined in Section 7.1. τ_{in} defines the bandwidth of the inner joint loop. ω_n is the natural frequency and ζ is the damping ratio of the second order system.

As the camera is static in this control stage, the tool pose cannot be recognized and measured visually if it is outside the camera range of view. To tackle this problem, we can estimate the 2D feature (image coordinates of the tool points) from the same model in Eqs. (72) and (73) with the joint angle q_m as input:

$$\tilde{u}_T = f_u \frac{Q(q_m) + \tan(\bar{q}_v)}{1 - Q(q_m) \tan(\bar{q}_v)} \quad (77)$$

where

$$Q(q_m) = \frac{L_t \sin(q_m)}{L - L_t \cos(q_m)} \quad (78)$$

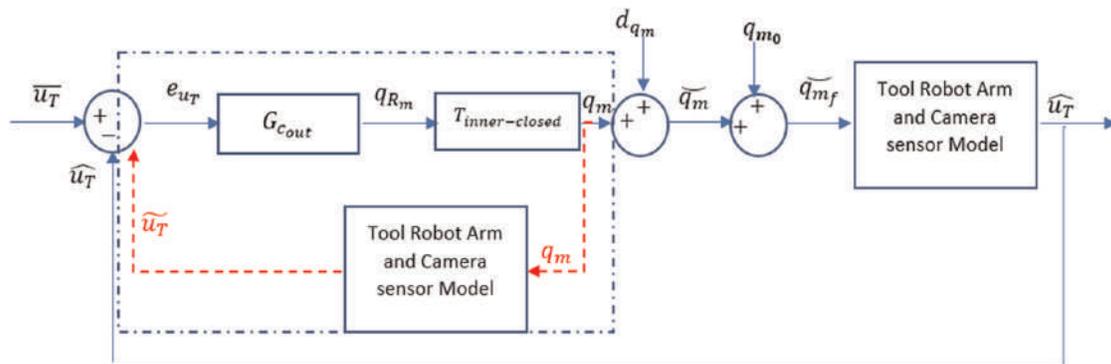


Figure 22.
 The block diagram of the tool manipulator feedback control loop with feature estimation.

which is illustrated in the block diagram of **Figure 22**. Normal feedback loop (in blue lines) is preserved when the tool is inside the camera range of view and hence, the camera can estimate the tool 2D feature \widehat{u}_T . However, when the tool is outside the range of view, the 2D feature can only be approximated as \widetilde{u}_T (red dashed line) by the combined model as shown in the blue dashed box. We can implement a bump-less switch to smoothly switch between these modes of operations. The switching signal changes over when the tool moves in or out of the camera range of view.

In addition, the feedforward controller, as shown in **Figure 8**, is designed with the inverse kinematics model of the tool robot arm and the details of this design is not provided here. It should be noted, as stated previously, that the combination of the feedforward and the feedback controllers provide a much faster response than the feedback controller by itself.

7.3 The simulation of cascaded SISO closed loop for high-accuracy tool manipulation control system

In this section, we present the simulation results of the feedforward and the feedback control system designed for the robot tool manipulator. As discussed, the plant linearization method is used in the feedback controller design. The controlled system is simulated with the original nonlinear tool robot arm and the camera sensor model of Eqs. (77) and (78). Furthermore, we illustrate how varying the damping ratio ζ affects the responses. We chose the bandwidth of the inner-loop control as 100rad/s and the bandwidth of the outer-loop control as 10rad/s . The intrinsic camera parameters are chosen as following: $f_u = 2.8\text{ mm}$ (Focal length) and $\alpha = 120^\circ$ (Angle of view). Other parameters are chosen as: $L = 1\text{m}$, and $L_t = 0.135\text{m}$.

We compare the performance of the feedforward-feedback control system and the feedback-only control system in two different scenarios. One scenario is simulated when the tool is kept inside the camera angle of view during the entire run time. It should be noted that both the feedback and the feedforward controllers are active during the entire simulation. The second scenario is simulated when the tool is outside the camera angle of view during the entire simulation time. When the tool pose is outside the camera angle of view, the feedback signal is replaced by the estimation signal from the model, as shown in **Figure 22**. Furthermore, for each scenario, we show how varying the damping ratio ζ affects the responses. Assuming the initial angle of the tool, $q_{m_0} = \mathbf{0}$ in the inertial base frame, we vary the pose or the rotational angle of the camera, \overline{q}_v , in the inertial base frame, for each simulation scenario.

Figures 23–26 show step responses of the joint angle \widetilde{q}_{mf} and the output tool pose P_{TX} (only X coordinate of the six dofs pose of the tool P_T) for the two scenarios. Each response is simulated with four different damping ratios: $\zeta=2$ (Blue), $\zeta = 1.5$ (Green), $\zeta = 1$ (Purple), and $\zeta = 0.7$ (Black). With the same damping ratio (same color), the response of the feedback-only control system is shown in the dashed line and the response of the feedforward-feedback control system is illustrated in the solid line.

In the first scenario, the camera rotates 5° counterclockwise with respect to the inertial frame. Then, the tool stays in the camera angle of view with any joint angle $\widetilde{q}_{mf} \in [-180^\circ, 180^\circ]$. In addition, a disturbance $d_{q_m} = -10^\circ$, is added to this joint angle. The responses in **Figures 23** and **24** illustrate that both the feedback-only and the feedforward-feedback control systems can reach stability and are robust to the disturbances. With varying the damping ratios, the feedforward-feedback system

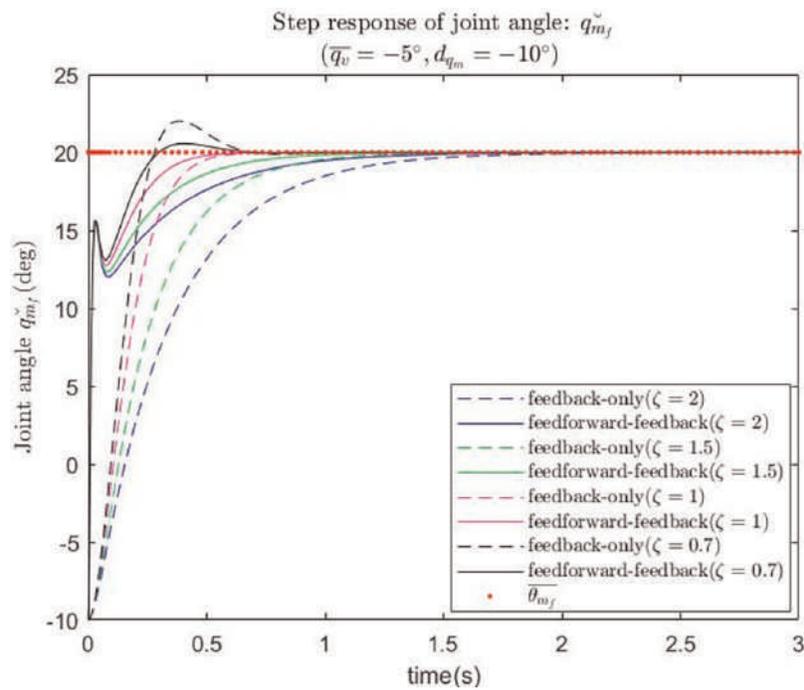


Figure 23.
 Step responses of \tilde{q}_{m_f} . Scenario 1: $\bar{q}_v = -5^\circ, d_{q_m} = -10^\circ$.

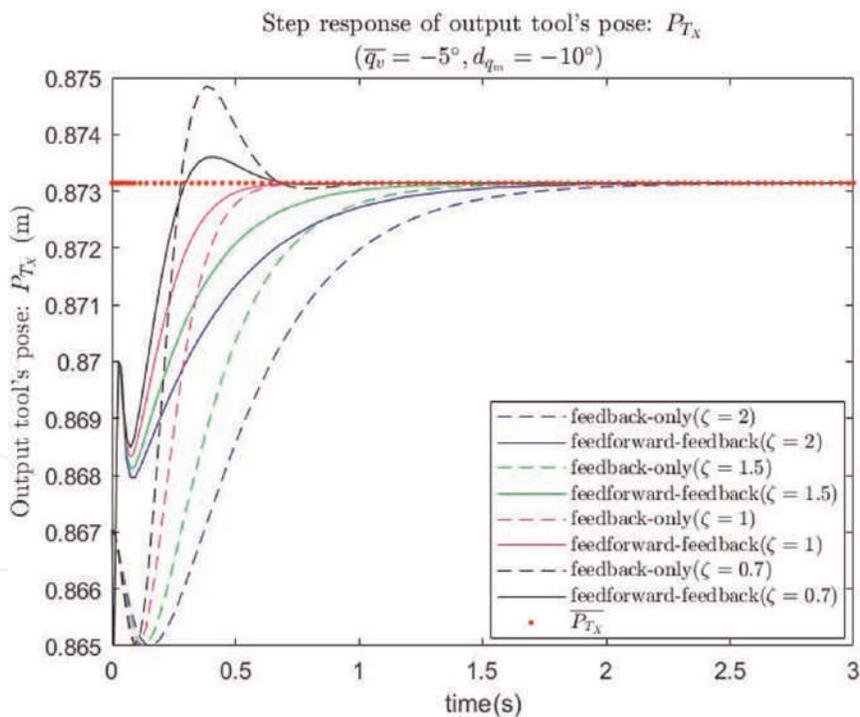


Figure 24.
 Step responses of P_{T_x} . Scenario 1: $\bar{q}_v = -5^\circ, d_{q_m} = -10^\circ$.

responses are faster in transient when compared to the response of the feedback-only system. The feedforward-feedback system response, when the damping ratio is small ($\zeta = 0.7$), results in a less overshoot when compared to the response of the feedback-only system. The optimal damping ratio, $\zeta_{opt} = 1$, (as discussed in Section 6.4), results in the fastest response and no overshoots, both for the feedforward-feedback and the feedback-only control systems.

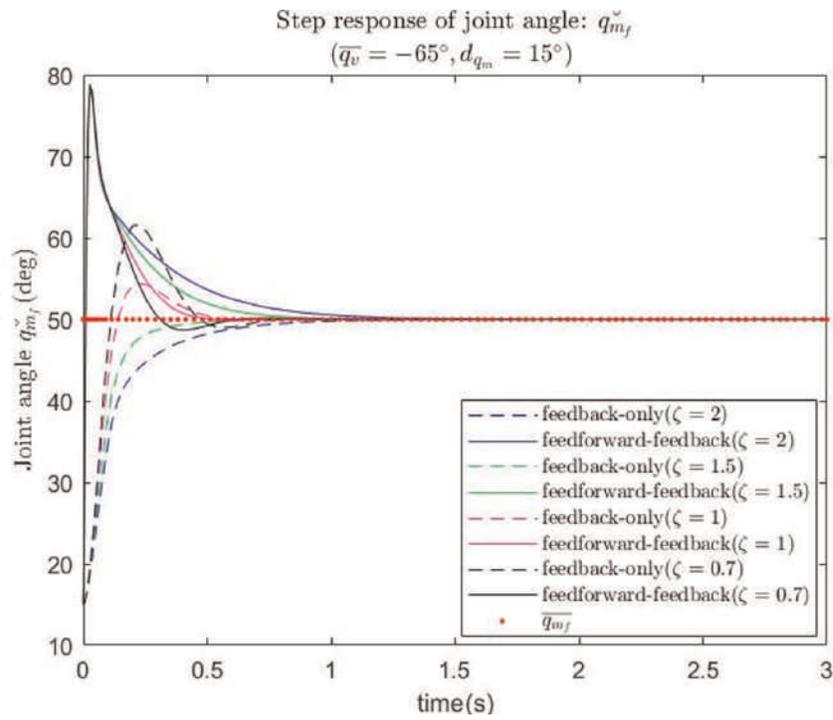


Figure 25.
 Step responses of \tilde{q}_{m_f} . Scenario 2: $\bar{q}_v = -65^\circ, d_{q_m} = 15^\circ$.

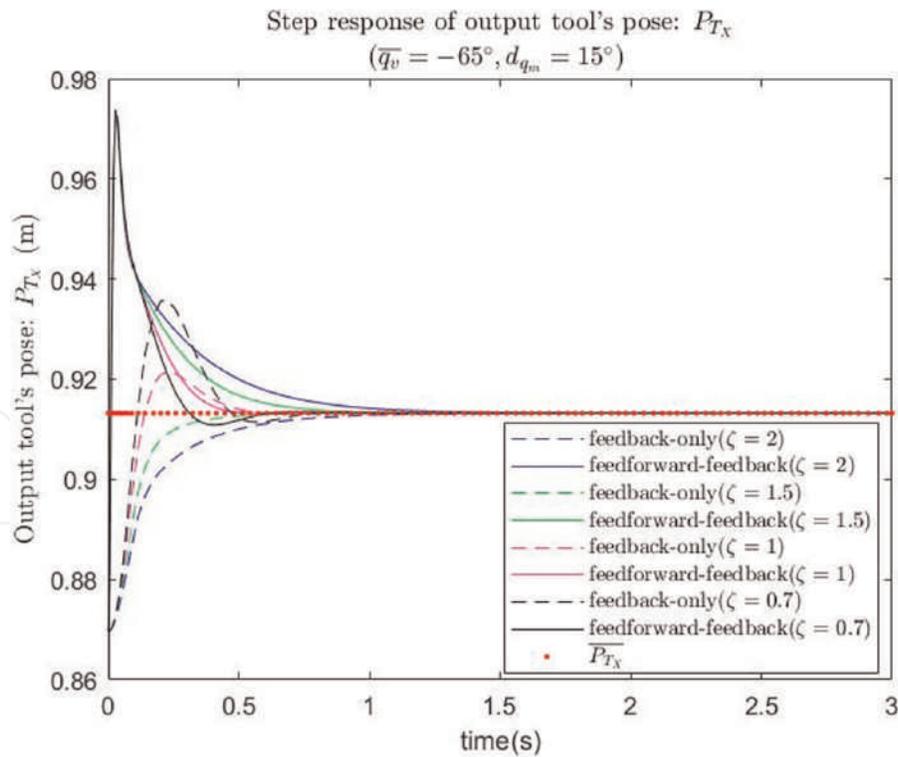


Figure 26.
 Step responses of P_{T_x} . Scenario 2: $\bar{q}_v = -65^\circ, d_{q_m} = 15^\circ$.

In the second scenario, the camera rotates 65° counterclockwise with respect to the inertial frame. Using geometry, we can calculate that the tool is out of the camera range of view when $\tilde{q}_{m_f} \notin [35.21^\circ, 134.79^\circ]$. Starting from the initial angle $q_{m0} = 0^\circ$

and move to the target joint angle $\overline{q_{mf}} = 50^\circ$, there is a range $\widetilde{q_{mf}} \in [0^\circ, 35.21^\circ]$ that camera cannot detect the tool but estimation of the pose is required to drive the tool to the target. Even perturbed with the disturbance $d_{q_m} = 15^\circ$, all the simulation responses shown in **Figures 25** and **26** reach the target within a second. In this scenario, the feedforward-feedback control system still converges faster in transient but generates bigger overshoots compared to the feedback-only control system. The large overshoots may come from accumulated disturbances that cannot be eliminated by the feedforward control without the intervention of the feedback control. A feedforward controller may drive the tool away from its target even faster when the disturbance appears in the loop. Perhaps, a possible solution, which will be investigated in the future, would be the use of a switching algorithm, switching from a feedforward to a feedback controller, rather than the use of a continuous feedforward-feedback controller.

In summary, the responses from a continuous feedforward-feedback system are more vulnerable to the disturbances especially when the starting position of the tool is far from its target. Although the disturbances will be eliminated as soon as the tool moves inside the camera range of view, the overshoots are more severe if more disturbances are accumulated in the process. In the real-world applications, the feedback-only control solutions are slower than the simulation results as the camera requires extra time, which is not considered in these simulations, to take pictures. Therefore, a feedforward controller, which compensates for the speed limitation of the feedback-only control, becomes indispensable in real manufacturing environments. The issue of the overshoots can be dealt with either by upgrading the camera with a wider range of view or as mentioned previously, the use of a switching algorithm, such as switching from a feedforward to a feedback controller, rather than the use of a continuous feedforward-feedback controller.

8. Conclusion

In this Chapter different sources of uncertainties in the task of positioning control in the automated manufacturing process are introduced. Then, a sequence of control methodologies is proposed. In the first part of this Chapter, we presented movement of a camera in the space to search for an optimal pose, a location in the space where the tool pose can be reached with minimum amount of energy and time duration. In the second part, we discussed a visual servoing architecture, which is applied to eliminate the measurement and dynamic noises occurred in the process of the camera movement. The image averaging technique is used to minimize the image noises by the averaging multiple images. In the last part, we designed the feedback and the feedforward controllers to guide the tool to its target by eliminating the dynamic errors in the tool movement process. Designs of all the control systems have been thoroughly discussed in this Chapter. Our methods for controller design are based on the classical Image Based Visual Servoing (IBVS) technique but are improved on by adding dynamic components to the systems and avoiding the depth estimation as done in the classical methods. Although only one degree of freedom case is discussed in this Chapter, the SISO simulation results have shown great potential of this work for various real-world applications in the automated high-speed manufacturing processes.

IntechOpen

IntechOpen

Author details

Rongfei Li* and Francis F. Assadian
Department of Mechanical Engineering, University of California, Davis, CA, USA

*Address all correspondence to: rfli@ucdavis.edu

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Wallén J. The history of the Industrial Robot. Technical Report 2853. Linköping, Sweden: Linköping University, Automatic Control; 2008
- [2] Conrad KL, Shiakolas PS, Yih TC. Robotic calibration issues: Accuracy, repeatability and calibration. In: Proceedings of the 8th Mediterranean Conference on Control and Automation (MED2000); 17–19 July 2000. Rio, Patras, Greece: Springer; 2000
- [3] Mironov D, Altamirano M, Zabihifar H, Liviniuk A, Liviniuk V, Tsetserukou D. Haptics of Screwing and Unscrewing for Its Application in Smart Factories for Disassembly. Cha Springer; 2018. DOI: 10.1007/978-3-319-93399-3_37
- [4] Anonymous. Robotic Accuracy Improves Aerospace Manufacturing. Industry Automation Review. 2020. Available from <https://industrialautomationreview.com/robotic-accuracy-improves-aerospace-manufacturing/> [Accessed: November 30, 2021]
- [5] Chaumette F, Hutchinson S. Visual servo control. I. Basic approaches. IEEE Robotics & Automation Magazine. 2006; 13(4):82-90. DOI: 10.1109/MRA.2006.250573
- [6] Hutchinson SA, Hager GD, Corke PI. A tutorial on visual servo control. IEEE Transactions on Robotics and Automation. 1996;12(5):651-670. DOI: 10.1109/70.538972
- [7] Chaumette F, Hutchinson S. Visual servoing and visual tracking. In: Siciliano B, Oussama K, editors. Handbook of Robotics. Berlin Heidelberg, Germany: Springer-Verlag; 2008. pp. 563-583. DOI: 10.1007/978-3-540-30301-5.ch25
- [8] Wilson WJ, Hulls CCW, Bell GS. Relative end-effector control using cartesian position based visual servoing. IEEE Transactions on Robotics and Automation. 1996;12(5):684-696. DOI: 10.1109/70.538974
- [9] Youla D, Jabr H, Bongiorno J. Modern wiener-Hopf Design of Optimal Controllers-Part II: The multivariable case. IEEE Transactions on Automatic Control. 1976;21(3):319-338. DOI: 10.1109/TAC.1976.1101223
- [10] Zhu W, Liu H, Ke Y. Sensor-based control using an image point and distance features for rivet-in-hole insertion. IEEE Transactions on Industrial Electronics. 2019;67(6):4692-4699. DOI: 10.1109/TIE.2019.2927186
- [11] Liu H, Zhu W, Ke Y. Pose alignment of aircraft structures with distance sensors and CCD cameras. Robotics and Computer-Integrated Manufacturing. 2017;48:30-38. DOI: 10.1016/j.rcim.2017.02.003
- [12] Liu H, Zhu W, Dong H, Ke Y. An adaptive ball-head positioning visual servoing method for aircraft digital assembly. Assembly Automation. 2019; 39(2):287-296. DOI: 10.1108/AA-05-2018-066
- [13] Du G, Wang K, Lian S, Zhao K. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review. Artificial Intelligence Review. 2021;54:1677-1734. DOI: 10.1007/s10462-020-09888-5
- [14] Avigal Y, Paradis S, Zhang H. 6-DoF grasp planning using fast 3D reconstruction and grasp quality CNN.

Machine Learning Papers. 2020; 8:1-6. DOI: 10.48550/arXiv.2009.08618.

[15] Wu C, Chen J, Cao Q, Zhang J, Tai Y, Sun L, et al. Grasp proposal networks: an end-to-end solution for visual learning of robotic grasps. *Advances in Neural Information Processing Systems*. 2020; 33:13174-13184. DOI: 10.48550/arXiv.2009.12606

[16] Li R, Platt R, Yuan W, Pas A, Roscup N, Srinivasan M, et al. Localization and manipulation of small parts using GelSight tactile sensing. *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014;20:3988-3993. DOI: 10.1109/IROS.2014.6943123

[17] Dong S, Rodriguez A. Tactile-based insertion for dense boxpacking. *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019;21:7953-7960. DOI: 10.1109/IROS40897.2019.8968204

[18] Hogan F, Ballester J, Dong S, Rodriguez A. Tactile dexterity: Manipulation primitives with tactile feedback. *The IEEE International Conference on Robotics and Automation (ICRA)*. 2020;23:8863-8869. DOI: 10.1109/ICRA40945.2020

[19] Fazeli N, Oller M, Wu J, Wu Z, Tenenbaum J, Rodriguez A. See, feel, act: hierarchical learning for complex manipulation skills with multisensory fusion. *Science Robotics*. 2019;4(26): eaav3123. DOI: 10.1126/scirobotics.aav3123

[20] Gregorio D, Zanella R, Palli G, Pirozzi S, Melchiorri C. Integration of robotic vision and tactile sensing for wire-terminal insertion tasks. *IEEE Transactions on Automation Science and Engineering*. 2019;16(2):585-598

[21] Flandin G, Chaumette F, Marchand E. Eye-in-hand/eye-to-hand cooperation for visual Servoing. *Proceedings—IEEE International Conference on Robotics and Automation*. 2000;3(3):2741-2746. DOI: 10.1109/ROBOT.2000.846442

[22] Hartley R, Zisserman A. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge: Cambridge University Press; 2004. DOI: 10.1017/CBO9780511811685

[23] Sonka M, Hlavac V, Boyle R. *Image Processing, Analysis, and Machine Vision*. 3rd ed. Pacific Grove: Brooks/Cole Publishing Company; 1999. pp. 373-421. DOI: 10.1007/978-1-4899-3216-.ch9

[24] Zhang Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000;22(11): 1330-1334. DOI: 10.1109/34.888718

[25] Tsai R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*. 1987;3(4): 323-344. DOI: 10.1109/JRA.1987.1087109

[26] Hartley. An algorithm for self-calibration from several views. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. London: IEEE; 1994. pp. 908-912. DOI: 10.1109/CVPR.1994.323923

[27] Luong QT, Faugeras O. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*. 1997;22:261-289. DOI: 10.1023/A:1007982716991

- [28] Mark WS, MV. Robot Dynamics and Control. Hoboken, New Jersey, United States: John Wiley & Sons, Inc.; 1989
- [29] Anonymous. ABB IRB 4600-40/2.55 Product Manual. 2013. Available from: <https://www.manualslib.com/manual/1449302/Abb-Irb-4600-40-2-55.html#manual> [Accessed: December 01, 2021]
- [30] Denavit J, Hartenberg RS. A kinematic notation for lower-pair mechanisms based on matrices. Transactions of the ASME, Journal of Applied Mechanics. 1955;23(2):215-221. DOI: 10.1115/1.4011045
- [31] Elatta AY, Gen LP, Zhi FL, Daoyuan Y, Fei L. An overview of robot calibration. Information Technology Journal. 2004;3:74-78. DOI: 10.3923/itj.2004.74.78
- [32] Renders J, Rossignol E, Becquet M, Hanus R. Kinematic calibration and geometrical parameter identification for robots. IEEE Transactions on Robotics and Automation. 1991;7(6):721-732. DOI: 10.1109/70.105381
- [33] Hayati SA. Robot arm geometric link parameter estimation. In: The 22nd IEEE Conference on Decision and Control. London: IEEE; 1983. pp. 1477-1483. DOI: 10.1109/CDC.1983.269783
- [34] Zhuang H, Wang L, Roth ZS. Simultaneous calibration of a robot and a hand-mounted camera. Proceedings IEEE International Conference on Robotics and Automation. 1993;2: 149-154. DOI: 10.1109/ROBOT.1993.292139
- [35] Wu C. A kinematic CAD tool for the design and control of a robot manipulator. The International Journal of Robotics Research. 1984;3(1):58-67. DOI: 10.1177/027836498400300105
- [36] Mooring BW, Roth ZS, Driels MR. Fundamental of Robotic Calibration. John Willey and Sons; 1991. pp. 221-225
- [37] Mooring BW, Tang GR. An improved method for identifying the kinematic parameters in a six axis robot. In: Proceedings of the ASME Computers in Engineering Conference, (ACES'83). Las Vegas, Nevada: ACES; 1984. pp. 79-84
- [38] Tsai RY, Lenz RK. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Transactions on Robotics and Automation. 1989;5(3):345-358. DOI: 10.1109/70.34770
- [39] Mooring BW, Padavala SS. The effect of kinematic model complexity on manipulator accuracy. Proceedings of International Conference on Robotics and Automation. 1989;1:593-598. DOI: 10.1109/ROBOT.1989.100049
- [40] Khalil W, Gautier M, Enguehard C. Identifiable parameters and optimum configurations for robots calibration. Robotica. 1991;9(1):63-70. DOI: 10.1017/S0263574700015575
- [41] Lee K, Shah DK. Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. IEEE Journal on Robotics and Automation. 1988;4(3): 361-367. DOI: 10.1109/56.797
- [42] Patidar P, Gupta M, Srivastava S, Nagawat AK. Image De-noising by various filters for different noise. International Journal of Computer Applications. 2010;9:45-50
- [43] Das S, Saikia J, Das S, Goñi N. A comparative study of different noise filtering techniques in digital images. International Journal of Engineering Research and General Science. 2015;3(5): 180-191

[44] Zhao R, Cui H. Improved threshold denoising method based on wavelet transform. In: 2015 7th International Conference on Modelling, Identification and Control (ICMIC). London: IEEE; 2015. pp. 1-4. DOI: 10.1109/ICMIC.2015.7409352

[45] Ng J, Goldberger JJ. Signal averaging for noise reduction. In: Goldberger J, Ng J, editors. Practical Signal and Image Processing in Clinical Cardiology. London: Springer; 2010. pp. 69-77. DOI: 10.1007/978-1-84882-515-4.ch7

[46] Fischler MA, Bolles RC. Random sample consensus: A Paradigm for model fitting with applications to image analysis and automated cartography. In: Fischler MA, Firschein O, editors. Readings in Computer Vision. San Francisco: Morgan Kaufmann; 1987. pp. 726-740. DOI: 10.1016/B978-0-08-051581-6.50070-2

[47] Chen G, Zhu F, Heng PA. An efficient statistical method for image noise level estimation. In: 2015 IEEE International Conference on Computer Vision (ICCV). London: IEEE; 2015. pp. 477-485. DOI: 10.1109/ICCV.2015.62

[48] Assadian F, Mallon K. Robust Control: Youla Parameterization Approach. Hoboken, New Jersey, USA: Jon Wiley & Sons, Inc.; 2022

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

171,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Robust Control Based on Input-Output Feedback Linearization for Induction Motor Drive: Real Time Implementation

Saber Krim and Mohamed Faouzi Mimouni

Abstract

This chapter proposes a design of hardware architecture of an improved Direct Torque Control (DTC) for a real-time implementation on a Xilinx Field-Programmable Gate Array (FPGA). The first contribution in this chapter consists in combining the DTC with a Space Vector Modulation (SVM) technique and an Input-Output Feedback Linearization (IOFL) approach. In fact, the classical DTC has remarkable performance in terms of fast torque response and less dependence on the system parameters. Despite the cited advantages, the classical DTC is penalized by high torque ripples and inverter-switching-frequency variations. In this context, the SVM is added to the DTC structure in order to keep the switching frequency constant and to reduce ripples. Furthermore, the nonlinear IOFL is proposed to achieve a decoupled flux and torque control. The novel structure is named in this chapter as DTC-IOFL-SVM. Moreover, this chapter presents a hardware implementation of the suggested DTC-IOFL-SVM strategy utilization. The hardware implementation is chosen in order to reduce the sampling period of the system thanks to the parallel processing of the FPGA. In order to demonstrate the performance of the FPGA implementation of the proposed DTC-IOFL-SVM, numerous simulation results are presented using the Xilinx system generator under a Matlab/Simulink.

Keywords: induction motor, direct torque control, input-output feedback linearization, FPGA

1. Introduction

Recently, the Direct Torque Control (DTC) of electrical machines has taken the attention of several researchers, thanks to its interest and advantages, like the simple structure, the fast torque response, and the less dependence on machine parameters [1–3]. The structure of the classical DTC is mainly based on two hysteresis controllers

and a lookup table to independently control the torque and the flux by selecting the optimal voltage vector in each sampling period. The classical DTC suffers from several problems like the torque ripples, the harmonics in the stator current waves, as well as the variation in the switching frequency. The fixed bands of the hysteresis controllers are the main cause of these problems [4, 5].

In the recent years, several methods have been put forward for overcoming the classical DTC problems, such as the use of intelligent techniques like the artificial neural networks and the fuzzy logic [6, 7]. However, the experimental implementation of the intelligent techniques requires powerful calculation processes due to their complexity. The torque and flux ripples, and the stator current waveform distortions can be reduced by fixing the switching frequency and selecting the more appropriate voltage vector for each commutation period. Indeed, in order to impose an operation with a fixed switching frequency, a combination between the DTC and the Space Vector Modulation (SVM) has been proposed by several research studies [8–10]. In fact, the DTC with a fixed switching frequency consists in introducing two Proportional Integral PI controllers and a SVM technique to achieve the best choice of the voltage vector in each sampling period [8–10]. However, the stability and dynamics of the system will be affected by the variation in machine parameters due to the existence of the PI controllers. In order to get rid of the drawbacks of the mentioned techniques, several robust control techniques have been proposed in order to guarantee the high performance control of induction motor drives. Among of these techniques we can cite the sliding mode control, the backstepping control and the Input–Output Feedback Linearization (IOFL) approach [11–13], which are the most popular control strategies. IOFL consists in transforming a nonlinear system into an equivalent linear one, which can be utilized for controlling the system [14]. IOFL is based on an inverse mathematical transformation for obtaining a suitable control law of the Induction Motor (IM).

The main first objective of this chapter consists in combining the IOFL technique and an SVM-DTC (SVM-DTC-IOFL) in order to design a novel DTC strategy featured by fast torque and speed responses, more robustness under stator resistance variations, reduced ripples and distortions, and a decoupled control between the torque and the flux. In this study, the stator flux and the electromagnetic torque are chosen as control states to develop the decoupled model of the IM.

For real time control of electrical machines, digital electronic boards like the STM32-microcontrollers [15, 16] and the Digital Signal Processor (DSP) are usually utilized [17–20]. The digital circuits based on microprocessors are known by their sequential computation of the control algorithm which consequently increases the execution time and the sampling period when the complexity of the control algorithm increases. Indeed, if the sampling time raises, the delays in the control system goes up, this causes additional ripples and distortions in the torque and the current, respectively. Moreover, the DSP controllers are chosen for implanting the control algorithms of electrical systems [21, 22], which are based on processor cores with high performance and few peripherals to communicate with the external environment. In fact, the sampling period of the processor depends of the computational burden due to the parallel processing, which creates delays in the feedback loop and raises the stator current harmonics and the torque ripples [23–25].

With the target of overcoming the DSP limitation and minimizing the DSP computational burden, a combination between the DSP and the FPGA has been proposed in the literature [26, 27] with the purpose of distributing the computational burden between these two digital controllers. This solution offers better performance by reducing the sampling period, the ripples in the torque and the distortions in the stator current.

However, the main limitations of this solution are the high cost and the complexity of circuit's connections, which causes problems for commercialization. In order to overcome the limitations of the cited solutions, the FPGA can be used only for controlling the motor drives. Indeed, thanks to its hardware architecture, the FPGA offers good performance by reducing the execution time and consequently the delays in the retroaction loop. In the last few years, the DSP (DSPACE 1104) has been suggested and confirmed by several engineers and researchers for real time control of AC machines [24, 28, 29]. In the same context, the FPGA can overcome the software solution drawbacks by adopting parallel processing [30–33]. In fact, the FPGA offers the designer the possibility of implementing in a low sampling period, control techniques with good performance and high algorithmic complexities. Indeed, in [32], the authors have implemented a control algorithm of an IM using an FPGA under a sampling period of 5 μ s [32].

The second objective of this chapter consists in implementing the proposed SVM-DTC-IOFL on an FPGA board. For the hardware implementation on the FPGA, the SVM-DTC-IOFL must be transformed into VHDL or Verilog description languages. Indeed, VHDL or Verilog programming is a difficult task which raises the design time, the time to market and the system cost. In this chapter, a graphical programming method based on Xilinx System Generator (XSG) is utilized in order to reduce the prototyping time. In fact, the graphical architecture from the XSG under a Matlab/Simulink-tool makes it possible to generate the VHDL of the Verilog code, as well as the programming bitstream files [33–35]. The XSG is a toolbox created by the Xilinx engineers' team, which operates between Matlab and Vivado tools, whose objective is to facilitate the programming tasks and reduce the time to market [35].

In this chapter, SVM-DTC-IOFL is theoretically developed, designed from the XSG tool, and verified by digital simulation utilizing a Xilinx Zynq FPGA.

This work is composed of five sections. In Section 2, the state mode of an induction motor drive, the SVM technique principle and the suggested IOFL theory are presented. In Section 3, designs from the XSG of the proposed SVM-DTC-IOFL and simulation results are shown. The implementation and synthesis results are given in Section 4. The conclusion is summarized in Section 5.

2. Theory and modeling

In this chapter, a combination between the SVM technique, the DTC strategy and the IOFL technique is put forward. The SVM is suggested in order to prevent ripples and distortions, and it provides an operation with a fixed switching frequency. IOFL is used in order to achieve decoupled control between the torque and flux quantities. The principle of these techniques is detailed in the following subsections.

2.1 Model presentation

The IM model is presented as follows, which will be used to design the proposed IOFL approach.

$$\dot{x} = f(x) + g v_{sa\beta} \quad (1)$$

with:

$$x = [i_{s\alpha} \ i_{s\beta} \ \phi_{s\alpha} \ \phi_{s\beta}]^T \quad (2)$$

$$f(x) = \begin{bmatrix} -\frac{1}{\sigma} \left(\frac{1}{T_r} + \frac{1}{T_s} \right) i_{s\alpha} - \omega_m i_{s\alpha} + \frac{1}{\sigma L_s T_r} \phi_{s\alpha} + \frac{\omega_m}{\sigma L_s} \phi_{s\beta} \\ \omega_m i_{s\alpha} - \frac{1}{\sigma} \left(\frac{1}{T_r} + \frac{1}{T_s} \right) i_{s\beta} - \frac{\omega_m}{\sigma L_s} \phi_{s\alpha} + \frac{1}{\sigma L_s T_r} \phi_{s\beta} \\ -R_s i_{s\alpha} \\ -R_s i_{s\beta} \end{bmatrix} \quad (3)$$

$$g = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 & 1 & 0 \\ 0 & \frac{1}{\sigma L_s} & 0 & 1 \end{bmatrix} \quad (4)$$

where:

$(i_{s\alpha}, i_{s\beta})$:the stator current components,

$(v_{s\alpha}, v_{s\beta})$:the voltage vectors components,

$(\phi_{s\alpha}, \phi_{s\beta})$:the stator flux vector components,

(R_r, R_s) : the rotor and stator resistance respectively,

(L_r, L_s) : the rotor and stator inductance respectively,

(T_r, T_s) : the rotor and stator time constants,

ω_m (red/sec): the electric rotor speed.

$\sigma = 1 - \frac{M_{sr}^2}{L_r L_s}$: The Blondel coefficient, where M_{sr} presents the mutual inductance.

2.2 Space vector modulation

The classical DTC based on fixed-bandwidth hysteresis controllers produces high ripples and distortions. Indeed, if a larger hysteresis-band of the torque is chosen, the torque ripples increase. For a smaller hysteresis band, the torque ripples are reduced and the switching frequency goes up, which consequently increases the commutation losses in the inverter IGBT transistors [36]. Thus, the SVM technique is proposed in this chapter in order to maintain a fixed switching frequency and reduce the ripples [37, 38]. The SVM principle consists in modulating reference voltage vector components in order to generate the more appropriate voltage vector that characterizes

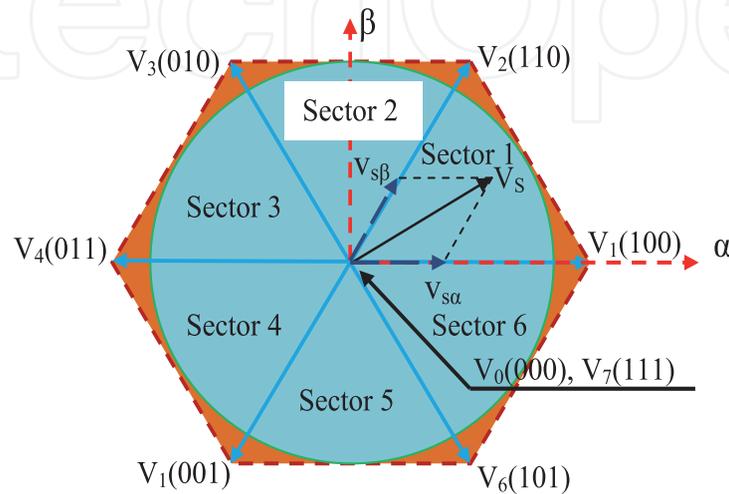


Figure 1.
Voltage vectors.

inverter control signals. As shown in **Figure 1**, the reference voltage vector can be determined by projecting it on the two vectors that bound the sector, using Eq. (5).

The time allowed for each voltage vector application can be determined by vector calculations. The rest of the sampling period can be filled by applying the null vector in order to grantee a fixed switching frequency [39, 40]. An example for the first section, by projection on vectors V_1 and V_2 , the voltage vector application times T_1 and T_2 are given by Eq. (5):

$$\left\{ \begin{array}{l} \vec{V}_S = V_{s\alpha}^* + jV_{s\beta}^* = \frac{T_1}{T_m} \vec{V}_1 + \frac{T_2}{T_m} \vec{V}_2 \\ \vec{V}_1 = \sqrt{\frac{2}{3}} U_{dc} (\cos(0) + j \sin(0)) = \sqrt{\frac{2}{3}} U_{dc} \\ \vec{V}_2 = \sqrt{\frac{2}{3}} U_{dc} \left(\cos\left(\frac{\pi}{3}\right) + j \sin\left(\frac{\pi}{3}\right) \right) \\ T_m = T_1 + T_2 + T_0 \\ T_1 = \left(\sqrt{\frac{3}{2}} V_{S\alpha} - \frac{1}{\sqrt{2}} V_{S\beta} \right) \frac{T_m}{U_{dc}} \\ T_2 = \sqrt{2} V_{S\alpha} \frac{T_{mod}}{U_{dc}} \end{array} \right. \quad (5)$$

where $(v_{s\alpha}^*, v_{s\beta}^*)$ represents the components of the reference voltage vector, T_1 and T_2 denote the commutation time, T_m is the sampling time, and U_{dc} is the DC voltage.

2.3 IOFL theory

This section illustrates the Feedback Linearization (FL) based DTC for an IM drive. The FL technique utilizes an inverse mathematical transformation in order to determine the desired control law for controlling the nonlinear system such as the IM. Furthermore, the FL technique is utilized to obtain decoupled control between the torque and flux. In this study, the suggested system outputs are the electromagnetic torque and the square root of the stator flux norm. Referring to the IOFL theory, the output variables are expressed as:

$$\left\{ \begin{array}{l} h_1(x) = T_{em} = \frac{3}{2} N_p (i_{s\beta} \phi_{s\alpha} - i_{s\alpha} \phi_{s\beta}) \\ h_2(x) = |\phi_s|^2 = \phi_{s\alpha}^2 + \phi_{s\beta}^2 \end{array} \right. \quad (6)$$

where T_{em} is the estimated electromagnetic torque, and $|\phi_s|$ is the norm of the stator flux. Assuming the controller objectives y_1 and y_2 as, we get:

$$\left\{ \begin{array}{l} y_1 = h_1(x) - T_{em}^* = T_{em} - T_{em}^* \\ y_2 = h_2(x) - |\phi_s^*|^2 = |\phi_s|^2 - |\phi_s^*|^2 \end{array} \right. \quad (7)$$

where T_{em}^* and $|\phi_s^*|$ are the torque and flux references, respectively. Utilizing the presented equations, the time derivative of the controller objectives can be written as:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} g_1(x) \\ g_2(x) \end{bmatrix} + G(x) \begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} \quad (8)$$

with:

$$\begin{cases} g_1(x) = \frac{3}{2}N_p \left[-\frac{1}{\sigma} \left(\frac{1}{T_r} + \frac{1}{T_s} \right) \phi_{s\alpha} i_{s\alpha} + \omega_m \phi_{s\alpha} i_{s\alpha} - \frac{\omega_m}{\sigma L_s} \phi_{s\alpha}^2 \right. \\ \left. + \frac{1}{\sigma} \left(\frac{1}{T_r} + \frac{1}{T_s} \right) \phi_{s\beta} i_{s\beta} + \omega_m \phi_{s\beta} i_{s\beta} - \frac{\omega_m}{\sigma L_s} \phi_{s\beta}^2 \right] - \dot{T}_{em}^* \\ g_2(x) = -2R_s \phi_{s\alpha} i_{s\alpha} - 2R_s \phi_{s\beta} i_{s\beta} - |\dot{\phi}_s^*| \\ G(x) = \begin{bmatrix} 2\phi_{s\alpha} & 2\phi_{s\beta} \\ \frac{3}{2}N_p \left(i_{s\beta} - \frac{1}{\sigma L_s} \phi_{s\beta} \right) & \frac{3}{2}N_p \left(i_{s\alpha} - \frac{1}{\sigma L_s} \phi_{s\alpha} \right) \end{bmatrix} \end{cases} \quad (9)$$

Based on the IOFL technique, the control inputs can be expressed as follows [41].

$$\begin{bmatrix} v_{s\alpha} \\ v_{s\beta} \end{bmatrix} = G^{-1}(x) \begin{bmatrix} -g_1(x) + v_1 \\ -g_2(x) + v_2 \end{bmatrix} \quad (10)$$

where v_1 and v_2 are assumed to be two auxiliary inputs with the purpose of ensuring more desired behavior and tracking accuracy for the torque and the stator flux, with:

$$\begin{cases} v_1 = -k_1 y_1 \\ v_2 = -k_2 y_2 \end{cases} \quad (11)$$

where k_1 and k_2 are positive constants. The SVM-DTC-IOFL performance strongly depends on the suitable choice of parameters k_1 and k_2 . In fact, the high values of such parameters are able to cause the system instability. On the other hand, the small values will lead to a poor robustness and slow convergence. Finally, it is necessary to better choose such parameters for guarantying high control technique performance [13]. The combination between (8), (10) and (11) gives the following expression:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 \\ 0 & -k_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (12)$$

Utilizing the IM model, the relation between the rotor and the stator fluxes is given below:

$$\begin{cases} \phi_{r\alpha} = \frac{\sigma L_s L_r}{M_{sr}} \left(\frac{1}{\sigma L_s} \phi_{s\alpha} - i_{s\alpha} \right) \\ \phi_{r\beta} = \frac{\sigma L_s L_r}{M_{sr}} \left(\frac{1}{\sigma L_s} \phi_{s\beta} - i_{s\beta} \right) \end{cases} \quad (13)$$

Utilizing matrix $G(x)$, defined in (9) and Eq. (13), the determinant of $G(x)$ is given as follows:

$$G(x) = \frac{3M_{sr}}{\sigma L_s L_r} N_p (\phi_{r\alpha} \phi_{s\alpha} + \phi_{r\beta} \phi_{s\beta}) \quad (14)$$

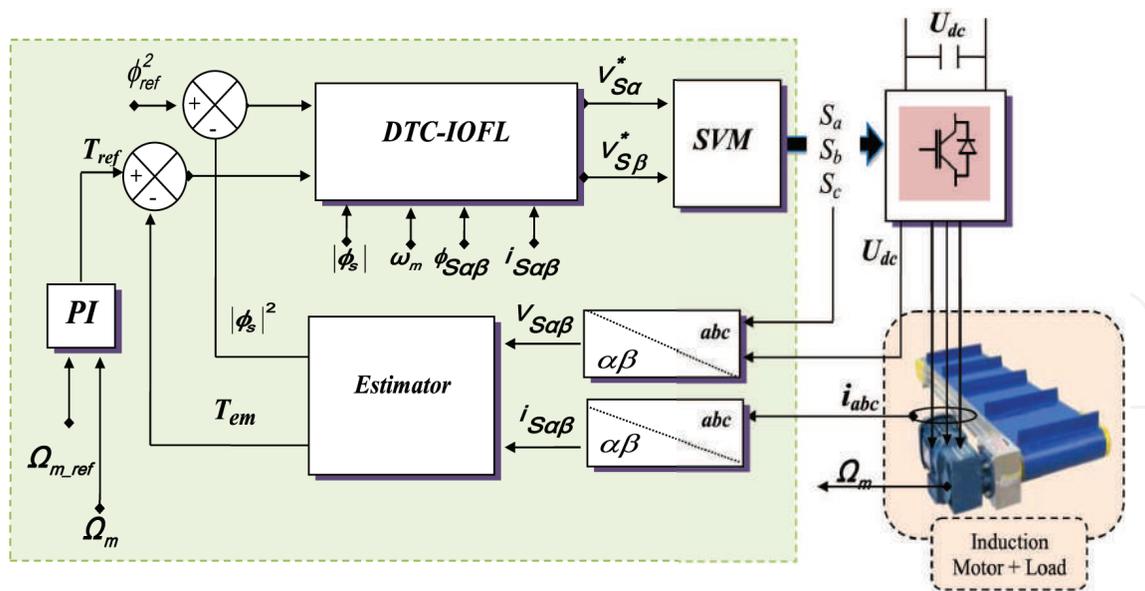


Figure 2.
 Global diagram of the proposed SVM-DTC-IOFL.

Referring to Eq. (14), it can be noticed that the product between the rotor flux and the stator flux cannot be zero, and matrix $G(x)$ is nonsingular [42].

The FL control law is used in order to satisfy the stability condition defined by the Lyapunov approach. To study the stability of the control law, the Lyapunov function is given as:

$$V = \frac{1}{2}y^T y \quad (15)$$

The time derivative of (15) is given as follows:

$$\dot{V} = y^T \dot{y} = [y_1 \quad y_2] \begin{bmatrix} -k_1 & 0 \\ 0 & -k_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = -k_1 y_1^2 - k_2 y_2^2 < 0 \quad (16)$$

Parameters k_1 and k_2 are positive, so derivative \dot{V} is negative, which demonstrates the stability of the control system. The global diagram of the proposed SVM-DTC-IOFL is given by **Figure 2**.

3. Simulation results and discussion

In this section, the simulation studies of an IM controlled by two control strategies, named classical DTC and SVM-DTC-IOFL, have been carried out under a Matlab/Simulink environment. The hardware architecture of the two control strategies are designed using XSG tool. The different parameters of the IM model are provided in **Table 1**.

The XSG tool is developed by Xilinx to be integrated into a Matlab/Simulink environment. It is widely utilized for the design, verification and implementation of control algorithms architectures on FPGAs. When we get the desired design with good

Parameter	Value	Parameter	Value
power (kW)	1.5	Rotor resistance (Ω)	4.282
Voltage (V)	230/400	Stator inductance (H)	0.464
Frequency (Hz)	50	Rotor inductance (H)	0.464
Pole pair	2	Mutual inductance (H)	0.4417
Stator resistance (Ω)	5.717	Rated speed (rpm)	1435

Table 1.
Induction machine parameters.

of simulation results, it will be possible for the XSG to automatically generate the VHDL code. As a matter of fact, the generated VHDL code will be used for generating the download. Bit file to be integrated into the FPGA. **Figure 3** depicts the design flow through the use of the XSG. **Figure 4** presents the SVM-DTC-IOFL architecture from the XSG.

3.1 First scenario

In this scenario, a comparative study between the classical DTC and the proposed SVM-DTC-IOFL is done under a rated load torque (10 Nm), a variable speed profile

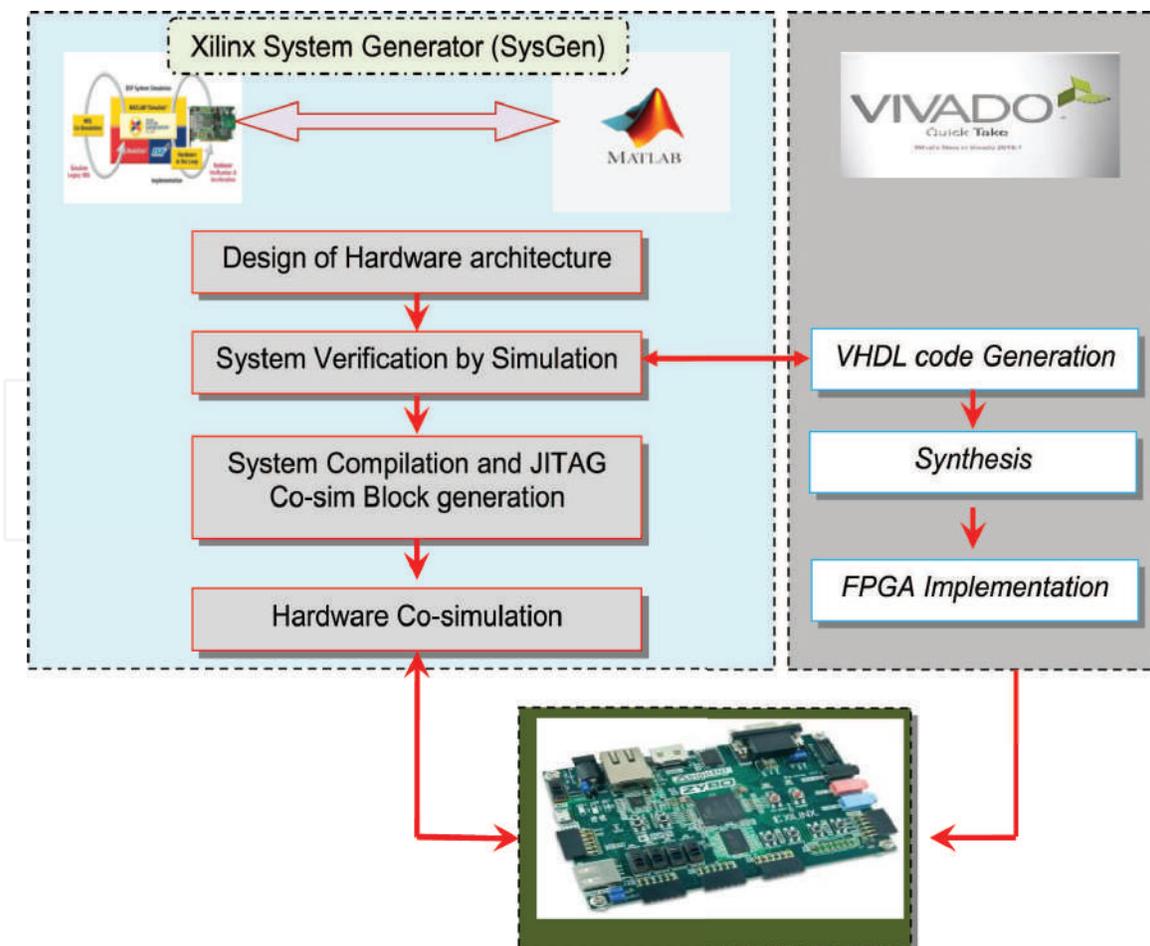


Figure 3.
XSG design flow.

and a reversal of the direction of rotation. In order to show the effectiveness of the suggested SVM-DTC-IOFL, it is compared with the classical DTC in terms of torque ripples and stator current distortion. The performance analysis is carried out with a sampling period equal to 100 μ s.

The IM starts with a reference speed equal to 100 rad/sec. At $t = 1$ sec the reference speed decreases slowly to reach -100 rad/sec at $t = 2$ sec. At $t = 0.5$ sec, a rated torque is applied.

Figure 5 presents the evolution of the rotor speed of the IM controlled by two control strategies. It can be noticed that the rotor speed converges to the reference speed for both control strategies. However, the proposed SVM-DTC-IOFL offers better performance in terms of ripples around the reference speed, as shown in **Figure 5(b)**. As given in **Figure 6(a)**, the suggested control strategy gives better performance in terms of ripples compared to the classical DTC (**Figure 6(b)**). **Figure 7** presents the three phase stator current consumed by the IM control by both control strategies. It can be seen that the proposed control strategy offers better performance in terms current distortions. In fact, for the suggested SVM-DTC-IOFL, the stator current has a smooth waveform (**Figure 7(a)**). **Figure 8** presents the evolution of the stator flux components for both control strategies. In can be seen that

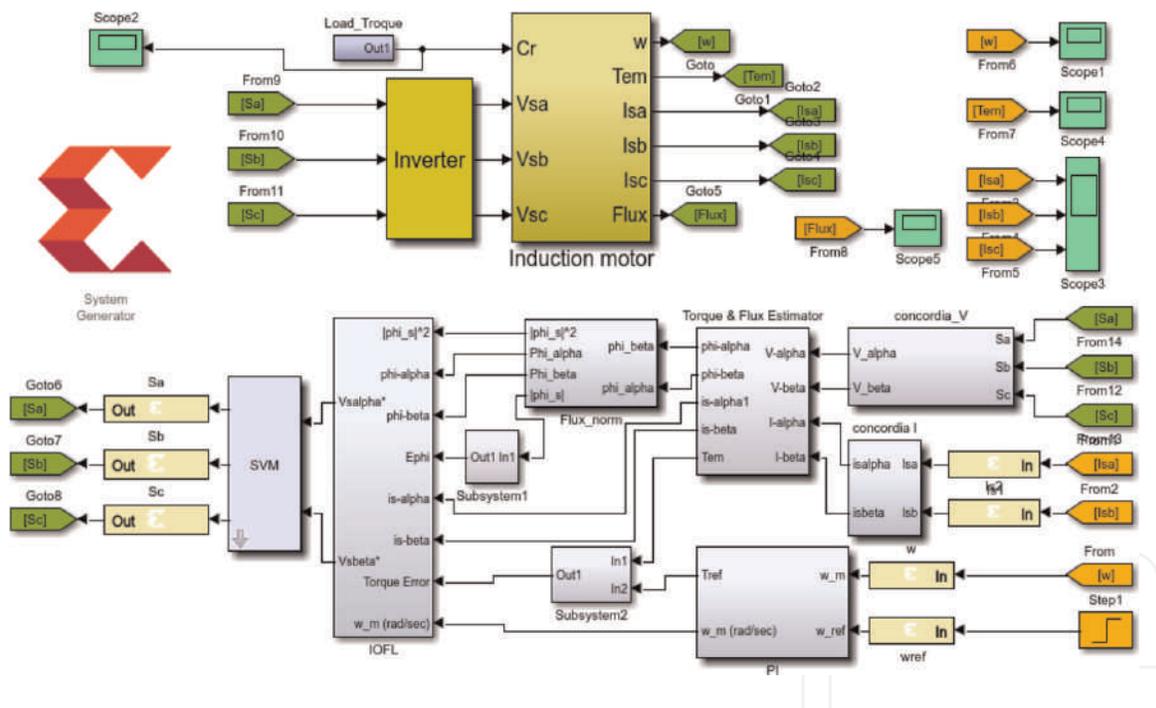


Figure 4. Full architecture of SVM-DTC-IOFL using XSG.

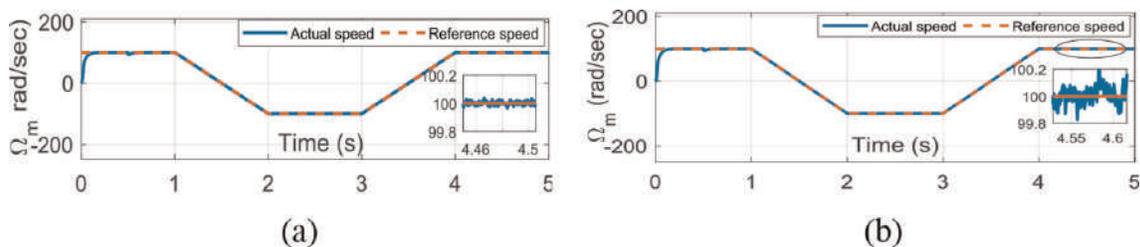


Figure 5. Speed response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

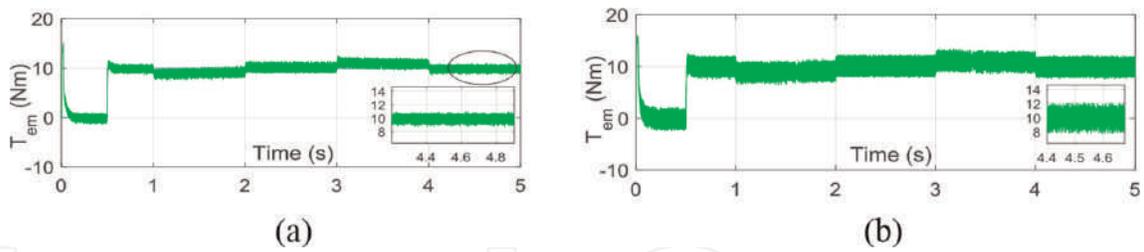


Figure 6. Torque response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

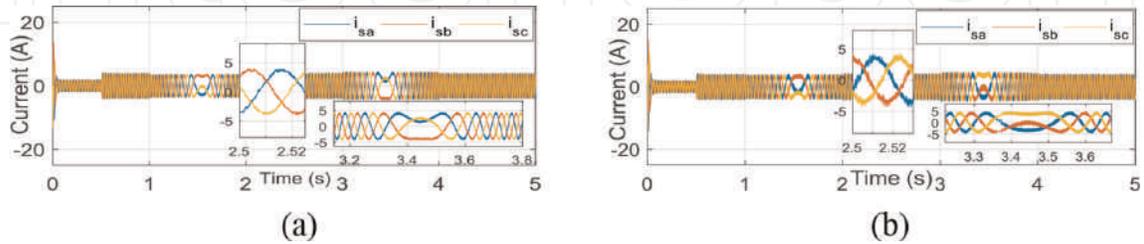


Figure 7. Three phase stator current for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

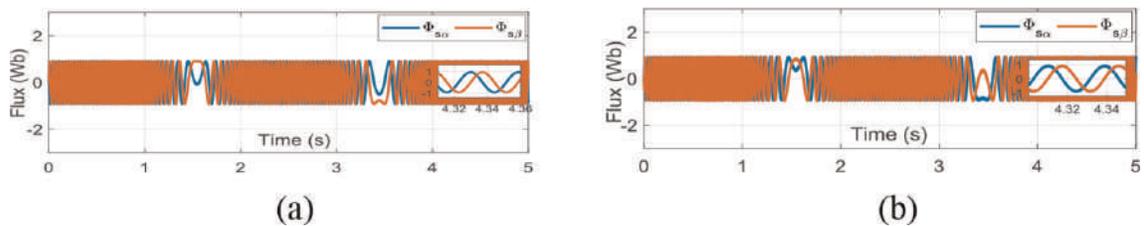


Figure 8. Three phase stator current for: (a) suggested SVM-DTC-IOFL, (b) classical DTC.

the real stator flux converges to its reference value. In addition, the proposed control strategy gives better performance in terms of flux-ripple reduction. More details are illustrated in **Table 2**.

3.2 Second scenario

In this scenario we used the same simulation conditions of the first scenario, but the main deference consists in reducing the sampling period which is equal to 10 μ s. In fact, when the control algorithm is implemented on software solutions like the

	Classical DTC	Proposed SVM-DTC-IOFL
Speed ripples	Medium	neglected
Torque ripples (%)	40%	20%
Current distortion	High	Medium
Sampling period	100 μ s	100 μ s

Table 2. Comparison between the both control strategies.

microcontrollers or the DSP, the sampling time increased due to the serial processing of these solutions, which consequently raises the control loop delay, the torque ripples and the stator current distortions. In order to overcome the limitations of these solutions in terms of execution time, the FPGA is proposed thanks to its parallel processing and short execution time. In order to show the effects of the execution time on the simulation results, a sampling period of 10 μ s is chosen. The obtained results in this scenario demonstrate that when the sampling period decreases, the torque and the stator flux ripples, as well as the stator current harmonics, are reduced, as shown in **Tables 2 and 3**.

The IM starts with a reference speed equal to 100 rad/sec. At $t = 1$ sec, the reference speed falls slowly to reach -100 rad/sec at $t = 2$ sec. At $t = 0.5$ sec, a rated torque is applied.

Figure 9 depicts the evolution of the rotor speed of the IM controlled by two control strategies. It can be noticed that the rotor speed converges to the reference speed for both control strategies. However, the suggested SVM-DTC-IOFL offers better performance in terms of ripples around the reference speed, as shown in **Figure 9(b)**. As given by **Figure 10(a)**, the proposed control strategy provides better performance in terms of ripples compared to the classical DTC (**Figure 10(b)**). **Figure 11** presents the three phase stator current consumed by the IM control by both control strategies. It can be seen that the suggested control strategy offers better performance in terms current distortions. In fact, for the proposed SVM-DTC-IOFL, the stator current has a smooth waveform (**Figure 11(a)**). **Figure 12** presents the evolution of the extremity of the stator flux vector in the Concordia reference. It can be noticed that when the motor is controlled by the classical DTC, the stator flux vector trajectory presents high deviations and ripples (as shown by **Figure 12(b)**). Contrariwise, in the case of the proposed SVM-DTC-IOFL a smooth circular trajectory is obtained as illustrated in **Figure 12(a)**. More details are given in **Table 3**.

	Classical DTC	Proposed SVM-DTC-IOFL
Speed ripples	Medium	neglected
Torque ripples (%)	10%	5%
Flux ripples (%)	4.39%	1.09%
Current distortion	High	neagleted
Sampling period	10 μ s	10 μ s

Table 3.
 Comparison between the both control strategies.

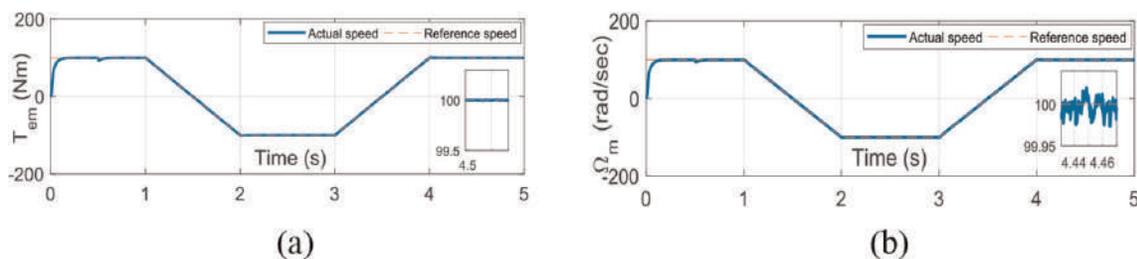


Figure 9.
 Speed response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

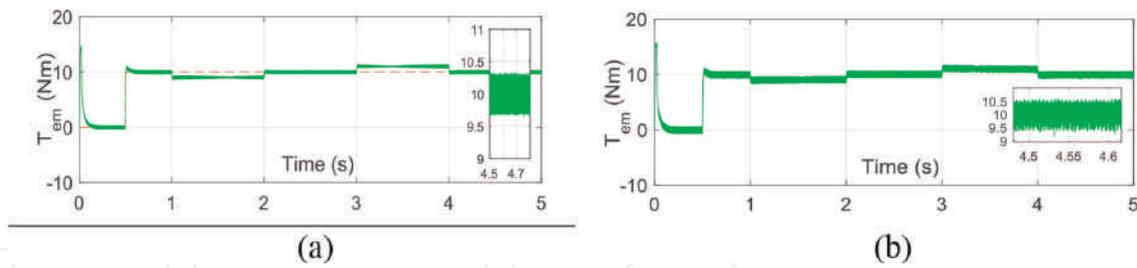


Figure 10. Torque response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

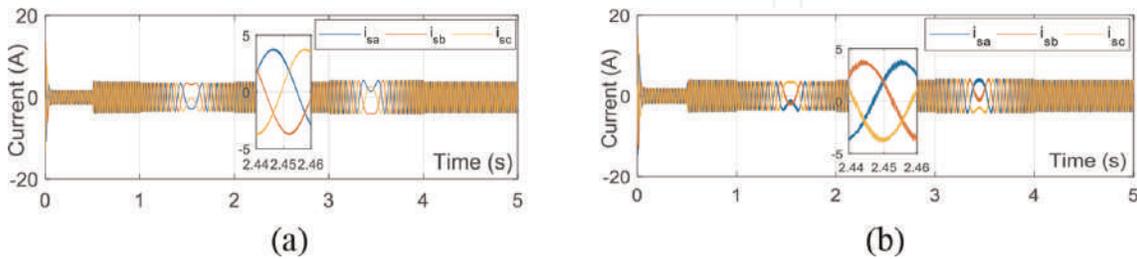


Figure 11. Three phase stator current for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

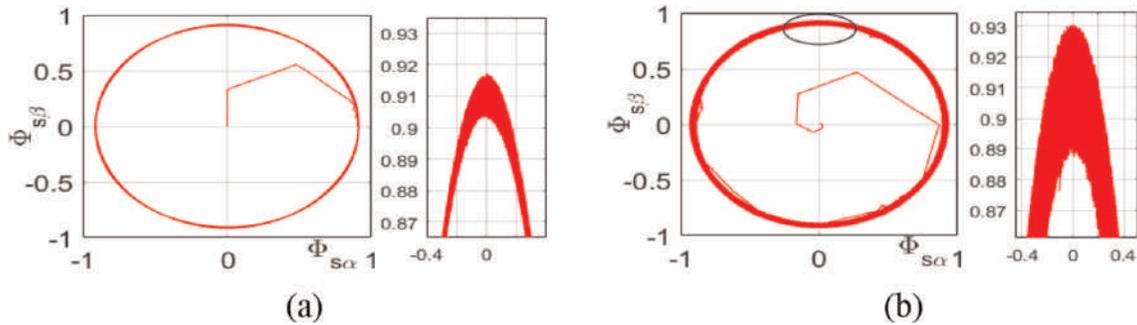


Figure 12. Three phase stator flux for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

3.3 Third scenario

This section consists in testing the robustness of the proposed SVM-DTC-IOFL under stator resistance variations at a low speed region. In this study, the IM starts with a reference speed equal to 20 rad/sec. The sampling period is equal to 10 μ s. At $t = 4$ sec, the stator resistance increases to reach 1.5 R_{sn} . **Figure 13(a, b)**, presents the evolution of the rotor speed for both control strategies. As shown in **Figure 13(a)**, it can be seen that the suggested SVM-DTC-IOFL offers better performance with a small deviation when the stator resistance goes up.

Figure 14(a, b) illustrates the evolution of the stator flux module for both control strategies. Referring to **Figure 14(a)**, it can be noticed that when the stator resistance rises, the stator flux curve presents small deviations and then it converges quickly to its reference value. However, when the IM is controlled by the classical DTC, the actual stator flux diverges from its reference value due to the variation in the stator resistance.

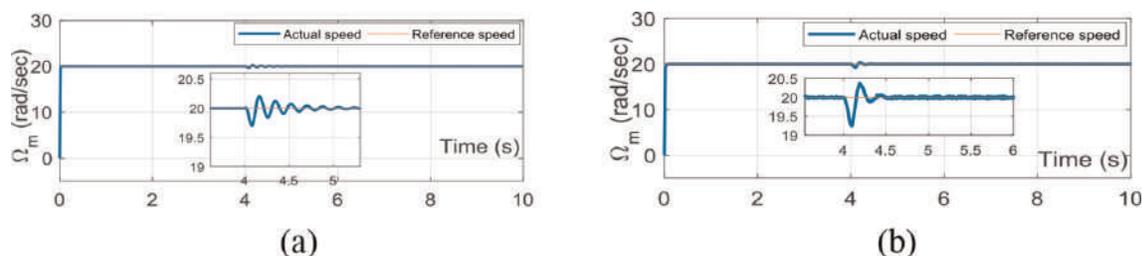


Figure 13.
 Speed response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

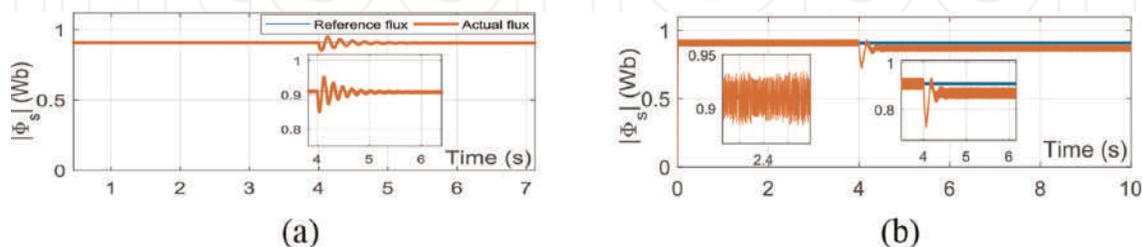


Figure 14.
 Speed response for: (a) proposed SVM-DTC-IOFL, (b) classical DTC.

4. VHDL code generation and synthesize results

The VHDL code generation and synthesis steps can be validated after verifying the functionality of the XSG architecture of the proposed SVM-DTC-IOFL. The obtained simulation results of the section confirm the good functionality of the designed XSG architecture, which offers the possibility to generate the VHDL and determine the synthesis results utilizing the Xilinx Vivado. During the hardware implementation of the classical DTC and the proposed SVM-DTC-IOFL approaches, the used resources from the FPGA are depicted in **Table 4**.

	Used with SVM-DTC-IOFL	Available
LUT	1436 (2.7%)	53,200
LUTRAM	104 (0.59%)	17,400
FF	2826 (2.65%)	106,400
BUFG	1 (3.12%)	32
DSP	10 (4%)	220

Table 4.
 Utilized resources from Xilinx Zynq FPGA.

5. Conclusion

In this chapter, a performance improvement of the DTC of an IM drive utilizing the SVM technique and a nonlinear control technique named IOFL has been presented. In order to solve the classical DTC problems, like the torque ripples, the current distortion and the variation in the switching frequency, the SVM has been

developed in this chapter. The proposed scheme is known as SVM-DTC. To increase the robustness of the suggested scheme under parameter variations, an IOFL approach has been combined with the SVM-DTC to generate the reference voltage vector. The real time implementation on the Xilinx Zynq FPGA has been put forward and investigated in this chapter so as to reduce the period of the system and eliminate the time delay in the control loop. The design of the proposed scheme has been carried out using the XSG toolbox. The flux and torque ripples have been considerably reduced thanks to the SVM technique. The nonlinear approach has given more performance, such as the robustness against the parameter variations, good and fast dynamic response and good tracking, and has reduced the complexity of the control scheme. Furthermore, the designed architecture of the control algorithm has been tested with two different sampling periods in order to demonstrate that if the sampling period rises, the ripples increase. Moreover, this controller has been featured by its simple design and implementation. The hardware FPGA implementation of the proposed SVM-DTC-IOFL can be considered as a good solution to control electrical motor drives.

For future work we are interested in the experimental validation of the proposed DTC-SVM-IOFL utilizing a real test bench.

Conflict of interest

The authors declare no conflict of interest.

Author details

Saber Krim^{1,2*} and Mohamed Faouzi Mimouni^{2,3}

1 Higher Institute of Applied Sciences and Technology of Kasserine, University of Kairouan, Kairouan, Tunisia

2 Laboratory of Automatic, Electrical Systems and Environment, National Engineering School of Monastir, University of Monastir, Tunisia

3 National Engineering School of Monastir, University of Monastir, Monastir, Tunisia

*Address all correspondence to: krim.saber.enim@gmail.com

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Casadei D, Profumo F, Serra G, Tani A. FOC and DTC: Two viable schemes for induction motors torque control. *IEEE Transactions on Power Electronics*. 2002;**17**:779-787
- [2] Ren Y, Zhu ZQ. Enhancement of steady-state performance indirect-torque- controlled dual three-phase permanent-magnet synchronous machine drives with modified switching table. *IEEE Transactions on Industrial Electronics*. 2015;**62**:3338-3350
- [3] Lai Y, Wang W, Chen Y. Novel switching techniques for reducing the speed ripple of AC drives with direct torque control. *IEEE Trans on Industrial Electronics*. 2004;**51**(4):768-775
- [4] Kang J-W, Sul SK. Analysis and prediction of inverter switching frequency in direct torque control of induction machine based on hysteresis bands and machine parameters. *IEEE Transactions on Industrial Electronics*. 2001;**48**:545-553
- [5] Idris NRN, Yatim AHM. Direct torque control of induction machines with constant switching frequency and reduced torque ripple. *IEEE Transactions on Industrial Electronics*. 2004;**51**(4)
- [6] Saberi H, Feyzi M, Sharifian MBB, Sabahi M. Improved sensorless direct torque control method using adaptive flux observer. *IET Power Electronics*. 2014;**7**:1675-1684
- [7] Fu X, Li S. A novel neural network vector control technique for induction motor drive. *IEEE Transactions on Energy Conversion*. 2015;**30**:1428-1437
- [8] Rashag HF, Tan NML, Koh SP, Abdalla AN, Chong KH, Tiong SK. DTC-SVM based on PI torque and PI flux controllers to achieve high performance of induction motor. *Research Journal of Applied Sciences, Engineering and Technology*. 2014;**7**(4):875-891
- [9] Joseline Metilda A, Arunadevi R, Ramesh N, Sharmeela C. Analysis of direct torque control using space vector modulation for three phase induction motor. *Recent Research in Science and Technology*. 2011;**3**(7):37-40
- [10] Saber KRIM, Gdaim S, Mtibaa A, Mimouni MF. Implementation on the FPGA of DTC-SVM based proportional integral and sliding mode controllers of an induction motor: A comparative study. *Journal of Circuits, Systems, and Computers*. 2017;**26**(3)
- [11] Zaafour A, Ben C, Ben AH, Châari A. DSP-based adaptive backstepping using the tracking errors for high-performance sensorless speed control of induction motor drive. *ISA Transactions*. 2015:1-15
- [12] Orłowska-Kowalska T, Tarchala G, Dybkowski M. Sliding-mode direct torque control and sliding-mode observer with a magnetizing reactance estimator for the field-weakening of the induction motor drive. *Mathematics and Computers in Simulation*. 2014:31-45
- [13] Alonge F, Cirrincione M, Pucci M, Sferlazza A. Input-output feedback linearization control with online MRAS-based inductor resistance estimation of linear induction motors including the dynamic end effects. *IEEE Transactions on Industry Applications*. 2016:254-266
- [14] Yazdanpanah R, Soltani J, Arab Markadeh GR. Nonlinear torque and stator flux controller for induction motor drive based on adaptive input-output feedback linearization and sliding mode

control. *Energy Conversion and Management*. 2008;**49**:541-550

[15] Motor Control With STM32®32-Bit ARM®-Based MCU. Pdf. Available from: http://www.st.com/web/en/resource/sales_and_marketing/promotional_material/brochure/brstm32mc.pdf

[16] STM32 embedded target for MATLAB and Simulink, Available from: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/data_brief/DM00080897.pdf

[17] Ozcira S, Bekiroglu N, Senol I. Dynamic performance and analysis of direct torque control method based on DSP for PMSM drives. In: *Proceedings of the IEEE International Conference on Renewable Energy Research and Applications*; Nagasaki, Japan. 2012. pp. 11-14

[18] Lianbing L, Hexu S, Xiaojun W, Yongqing T. A high-performance direct torque control based on DSP in permanent magnet synchronous motor drive. In: *Proceedings of the IEEE 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527)*; Shanghai, China. Vol. 2. 10-14 June 2002. pp. 1622-1625

[19] Cruz SMA et al. DSP implementation of the multiple reference frames theory for the diagnosis of stator faults in a DTC induction motor drive. *IEEE Transactions on Energy Conversion*. 2005;**20**:329-335

[20] Singh B, Singh BP, Dwivedi S. DSP based implementation of sliding mode speed controller for direct torque controlled PMSM drive. In: *Proceedings of the IEEE International Conference on Industrial Technology*; Mumbai, India. 2006. pp. 1301-1308

[21] Monmasson E, Cirstea MN. FPGA design methodology for industrial

control systems—A review. *IEEE Transactions on Industrial Electronics*. 2007;**54**(4):1824-1842

[22] Cecati C. Microprocessors for power electronics and electrical drives applications. *IEEE Ind. Electron. Soc. Newslett.* 1999;**46**(3):5-9

[23] Hussein F, Malik E. Improving the torque ripple in DTC of PMSM using fuzzy logic. In: *IEEE Industry Applications Society Annual Meeting*; Edmonton, AB, Canada. 5-9 October 2008. pp. 1-8. DOI: 10.1109/08IAS.2008.162

[24] Idkhajine L, Monmasson E, Naouar MW, Prata A, Bouallaga K. Fully integrated FPGA-based controller for synchronous motor drive. *IEEE Transactions on Industrial Electronics*. 2009;**56**(10)

[25] Lokriti A, IssamSalhi SD. IM direct torque control with no flux distortion and no static torque error. *ISA Transactions*. 2015;**59**:256-267

[26] Idris NRN, Chuen LT, Elbuluk ME. A new torque and flux controller for direct torque control of induction machines. *IEEE Transactions on Industry Applications*. 2006;**42**(6):1358-1366

[27] Jidin A, Idris NRN, Yatim AHM, Sutikno T, Elbuluk ME. An optimized switching strategy for quick dynamic torque control in DTC-hysteresis-based induction machines. *IEEE Transactions on Industrial Electronics*. 2011;**58**(8): 3391-3400

[28] Tabbache B, Benbouzid M, Kheloui A, Bourgeot JM, Mamoune A. An improved fault-tolerant control scheme for PWM inverter-fed induction motor-based EVs. *ISA Transactions*. 2013;**52**:862-869

[29] Rafa S, Larabi A, Barazane L, Manceur M, Essounbouli N,

- Hamzaoui A. Implementation of a new fuzzy vector control of induction motor. *ISA Transactions*. 2014;**52**:744-754
- [30] Shahbazi M, Poure P, Saadate S, Zolghadri MR. FPGA-based reconfigurable control for fault-tolerant Back-to-Back converter without redundancy. *IEEE Transactions on Industrial Electronics*. 2013;**60**(8)
- [31] Jezernik K, Korelic J, Horvat R. PMSM sliding mode FPGA based control for torque ripple reduction. *IEEE Transactions on Power Electronics*. 2013;**28**(7)
- [32] Sutikno T, Idris NR, Jidin A, Cirstea MN. An improved FPGA implementation of direct torque control for induction machines. *IEEE Transactions on Industrial Informatics*. 2013;**9**(3)
- [33] Gafsi M, Abbassi N, Hajjaji MA, Malek J, Mtibaa A. Xilinx Zynq FPGA for hardware implementation of a chaos-based cryptosystem for real-time image protection. *Journal of Circuits, Systems and Computers*. 2021;**30**(11): 2150204
- [34] Saidani T, Atri M, Dia D, Tourki R. Using Xilinx system generator for real time hardware Co-simulation of video processing system. *Electronic Engineering and Computing Technology*. 2010;**60**:227-236
- [35] Xilinx System Generator User's Guide. UG640 (v11.4), 2009. Available from: http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/sysgen_user.pdf
- [36] Uddin M, Hafeez M. FLC-based DTC scheme to improve the dynamic performance of an IM drive. *IEEE Transactions on Industry Applications*. 2012;**48**:823-831. DOI: 10.1109/TIA.2011.2181287
- [37] Bounadja M, Belarbi A, Belmadani B. A high performance space vector modulation–direct torque controlled induction machine drive based on stator flux orientation technique. *Adv. Electr. Comput. Eng.* 2009;**9**(2):28-33
- [38] Rashag HF, Koh SP, Chong KH, Tiong SK, Tan NML, Abdalla AN. High performance of space vector modulation direct torque control SVM-DTC based on amplitude voltage and stator flux angle. *Research Journal of Applied Sciences, Engineering and Technology*. 2013;**5**(15): 3934-3940
- [39] Habetler TG, Profumo F, Pastorelli M, Tolbert LM. Direct torque control of induction machines using space vector modulation. *Ind Appl IEEE Trans*. 1992;**28**:1045-1053. DOI: 10.1109/28.158828
- [40] Ammar A, Bourek A, Benakcha A. Modified load angle direct torque control for sensorless induction motor using sliding mode flux observer. In: *Proceedings of the IEEE 4th Conference on Electrical Engineering; Boumerdes, Algeria*. 13-15 December 2015. pp. 1-6
- [41] Soltani J, Markadeh GRA, Abjadi NR, Ping HW. A new adaptive direct torque control (DTC) scheme based-on SVM for adjustable speed sensorless induction motor drive. In: *Proceedings of the IEEE International Conference on Electrical Machines and Systems (ICEMS); Seoul, Korea (South)*. 8-11 October 2007. pp. 1111-1116
- [42] Zhang Z, Tang R, Bai B, Xie D. Novel direct torque control based on space vector modulation with adaptive stator flux observer for induction motors. *IEEE Transactions on Magnetics*. 2010;**46**: 3133-3136