HIGH-PERFORMANCE CONTROL OF IPMSM BY USING MODIFIED-DTC ALGORITHM AND ANN BASED SPEED CONTROLLER

FOR THE DEGREE OF

MASTER PHILOSOPHY

BY

JIA ZHENYU

School of Electrical Engineering

Graduate School of the University of Ulsan

May, 2018

HIGH-PERFORMANCE CONTROL OF IPMSM BY USING MODIFIED-DTC ALGORITHM AND ANN BASED SPEED CONTROLLER

A THESIS

SUBMITTED

FOR THE DEGREE OF

MASTER PHILOSOPHY

BY

JIA ZHENYU

UNDER THE SUPERVISION OF

Prof. Byeongwoo Kim

School of Electrical Engineering

Graduate School of the University of Ulsan

May, 2018

HIGH-PERFORMANCE CONTROL OF IPMSM BY USING MODIFIED-DTC ALGORITHM AND ANN BASED SPEED CONTROLLER

Approved by:

Prof.
School of Electrical Engineering (Signature)
University of Ulsan

Prof.
School of Electrical Engineering (Signature)
University of Ulsan

PhD.
Korea Automotive Technology Institute (Signature)

Date Approved: May, 2018

ACKNOWLEDGMENTS

During the master course，I experienced and learned a lot of things. I would like to thank many people who have contributed to my research work of this thesis and life in University of Ulsan.

I express my sincere gratitude towards my advisor Professor Byeongwoo, Kim. He is kind and patient all the time. His trust, care and encouragement throughout the past years helped me to insist on researching and solving problems. It is very hard for me to do research work at the beginning, but now I feel honorable for being here. Without his support and patience, my research work and the completion of this thesis would have never become possible.

I would also like to give my heartfelt thanks to the professor Heejun Kang and Hansil, Kim. They are nice and trusted professors. What's more, they gave me the opportunity to develop my knowledge of robot control and linear system control, which are important and helpful for me to study servo system control.

Moreover, I give my thankfulness to professor Kanghyun, Jo. In his lectures, I improved my academic English skills and started to know and learn artificial intelligence technique, which is hot popular and very powerful subject. Through his lectures, I also learned about image detection theory, which is important for autonomous vehicle development.

I thank all my colleagues of Automotive Electronics & Control Laboratory, they helped me in my daily lab life and gave me guidance of living in South Korea. Lin Ming helped me a lot in language translation and information delivering. Jungeun Lee is always helpful to me when I have troubles in lab life, I appreciate it deeply. I am grateful to my friend, Wang He, for his helps in paper writing skills.

Last but not the least, I acknowledge the grace of almighty Jesus who leads me at any time and everywhere, Soli Deo Gloria. I sincerely thank my family for the supports and encouragements.

This research was supported by the Industry Core Technology Development Project "Development of 2 *kw* / *kg* ,100 *kw* class IPMSM electric drive system with high efficiency cooling" through the Ministry of Trade, Industry and Energy (Grant Number: 10063006).

SUMMARY

HIGH-PERFORMANCE CONTROL OF IPMSM BY USING MODIFIED-DTC ALGORITHM AND ANN BASED SPEED CONTROLLER

Zhenyu Jia

School of Electrical Engineering

The Graduate School

University of Ulsan

In this thesis, we present a performance enhancement on direct torque control (DTC) for interior permanent magnet synchronous motor (IPMSM) drive. To solve some key drawbacks of the conventional DTC and space vector modulation (SVM) based DTC control strategies, the predictive controller with super-twist sliding mode (STSM) based torque controller is applied to modify SVM based DTC and backpropagation algorithm based neural network is adopted to tune the parameters of speed controller.

First, a novel STSM torque controller based predictive calculator for generating reference voltage vector is analysed and designed in Chapter 2. This modified DTC strategy is named as STSM-DTC in this thesis. In previous research [27], the classical proportional-integral (PI) torque controller is applied and named as PI-DTC method. In this thesis, both STSM-DTC and PI-DTC strategies combine SVM scheme applied to IPMSM drive. Generally, the torque-ripple and flux-ripple reduction performance has been improved considerably by adopting SVM. The simulation results demonstrate that the proposed STSM-DTC method can enhance the torque control performance furtherly more compared to PI-DTC with smaller torque ripple. Moreover, it has the merits of fast dynamic response, robustness

against load torque disturbance and speed-ripple reduction, as verified in Chapter 4. In addition, the whole proposed STSM-DTC control strategy, for IPMSM drives can obtain good performance of stator current with very low THD (total harmonic distortion) level.

Next, a novel PI speed controller applying backpropagation based neural network is designed and proposed in Chapter 3. This speed controller is named as NN-PI speed controller in this thesis. The proposed method can overcome the disadvantages of conventional PI controller with high dependence on controlled model and time-consuming parameters-tuning work. Compared to classical PI speed controller, the simulation results prove the effectiveness of the proposed NN-PI speed controller with faster dynamic response, good speed tracking, much smaller speed overshoot and better robustness against load torque variation and load inertia variation, as presented in Chapter 4.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background and Objectives

In recent years, the research and development on Electric Vehicles/ Hybrid Electric Vehicles (EVs/ HEVs) has become extensively popular due to the serious air pollution and the shortage of energy resources. For the electric propulsion systems in EVs/HEVs, the induction machine (IM), permanent magnet synchronous motor (PMSM), and switched reluctance motor (SRM)are usually considered [1]. However, owing to its outstanding advantages like high power density, high efficiency, simple structure and low maintenance cost, the PMSMs are extremely applied [2].

Inherent coupled flux and torque is the main disadvantage of PMSMs, which makes them hard to control. Many advanced control techniques have been presented to make PMSMs more comparable. Filed-oriented control (FOC) [3] and direct torque control (DTC) [4] are the two main control techniques for PMSM drives. The FOC technique is widely used in PMSM drives. However, it performs not as well as predicted in practical engineering application due to the variations of motor parameters and inaccurate control mode [5]. In 1980s, DTC algorithm [6] was firstly introduced from ABB for induction motor drives. Decades past, it was already developed and used for variety of motor drives.

Compared with FOC, the conventional DTC is a significantly new concept with capability of controlling the electromagnetic torque and stator flux linkage directly. It has many characteristics of less dependence on machine parameters, fast dynamic response and less complexity without coordinate transformation or current regulation, and robustness against motor parameters' variation and external disturbances [7]. Furthermore, it doesn't need extra sensors to implement DTC. Hence, the DTC surpasses FOC in the application of EVs. However, as reported in literatures [2], [4], [7-12], this technique still possesses several major disadvantages, namely relatively large torque and flux ripples, variable switching frequency and high sampling requirement for digital implementation. Moreover, conventional DTC usually adopts a PI speed controller in the outer loop. Consequently, it usually affects

the dynamic and stability of the system like improper speed response with high overshoot at the motor start stage, big speed ripples when system variations occur [11].

To solve the aforementioned-problems of conventional DTC, varieties of modified DTC schemes have been presented from various perspectives. One category is to add more available voltage vectors by using new hardware topologies. In [8], a multilevel convertor is employed to generate more voltage vectors for reducing torque ripples. However, it needs more power switches thus increasing the system cost and complexity. The category of nonlinear control technique was also presented. In [9-10], feedback linearization control (FLC) is used to convert the coupled nonlinear PMSM control system to an equivalent linear one. The FLC based DTC scheme in [10] achieves fast torque response and small torque and flux ripples. In these nonlinear FLC strategies, many motor parameters are used to get state variables. Hence, the robustness of the system will be deteriorated when parameters vary. Another category to enhance DTC scheme is the employment of space vector modulation (SVM) [2], [11], [13]. SVM based DTC can obtain voltage vectors with adjustable amplitude and phase, thus controlling the electromagnetic torque and stator flux more accurately. By incorporation with SVM, the DTC can be improved with torque and flux ripples reduction and fixed switching frequency. The fundamental issue of SVM based DTC is to get the reference voltage vectors. Some methods to obtain reference voltage vectors have been proposed in the literatures [2], [12-13].

All the strategies mentioned above are presented to improve the torque control of DTC controlled motors, but neglecting speed control which also plays an important role in electric machines. Generally, PI controller is widely applied in the outer speed regulator loop due to its relatively simple implementation and effectiveness [14]. However, the PI control approach cannot perform sufficiently well in nonlinear PMSM drives system with various uncertainties. Moreover, it's time consuming and hard to tune the PI controller parameters which depends on the system [15]. Recently, some new methods have been proposed to replace PI speed controller. In [2], [9], sliding mode control (SMC) are applied in speed controller owing to its great merits such as robustness to external load disturbance and fast dynamic response. However, its system dynamics are still affected by parameter variations and chattering problems [18]. Meanwhile, adaptive speed control schemes are also proposed to control the

speed of PMSM [17-18]. With capability of automatic adjustment, adaptive control is the trend of digital control, but it's hard to master the adaptive control law [15]. Next, the artificial neural network (ANN) is also an interesting scheme in the speed control of PMSM, due to its ability of approximate the linear or nonlinear mapping through learning. In [16], neural speed controller is designed and adopted to improve the control performances of vector control based PMSM. Nevertheless, only load torque variations and speed step change are considered.

In this thesis, a novel SVM based modified DTC algorithm by using predictive controller with super twisting sliding mode control (STSM) is adopted for electromagnetic torque and stator flux ripples reduction and fixed switching frequency, named as STSM-DTC method. Moreover, a novel adaptive speed controller based on neural network PI (NN-PI) is designed instead of a classical PI controller. Consequently, simulation results demonstrate that proposed NN-PI speed controller combining STSM-DTC strategy can obtain high performance with fast dynamic response, small torque and flux ripples, small speed overshoot and speed drop against load torque and load inertia disturbance.

1.2 The mathematical modeling of IPMSM

This sub-chapter describes the basic mathematical modeling of PMSM in different reference coordinate.

(1) Modeling in $d-q$ rotating frame

It's convenient to design the control scheme when presenting IPMSM model in $d-q$ rotating frame where d-axis coordinates with rotor at same speed. The equations of IPMSM in the rotating frame are shown as followings:

Stator voltage in $d-q$ axis:

$$\begin{cases} u_d = Ri_d + \dfrac{d}{dt}\psi_d - p_n\omega_r\psi_q \\ u_q = Ri_q + \dfrac{d}{dt}\psi_q - p_n\omega_r\psi_d \end{cases} \tag{1-1}$$

Stator flux in $d-q$ axis:

$$\begin{cases} \psi_d = L_d i_d + \psi_f \\ \psi_q = L_q i_q \end{cases} \tag{1-2}$$

Electromagnetic torque:

$$T_e = \frac{2}{3} p_n i_d \left[i_d \left(L_d - L_q \right) + \psi_f \right] \tag{1-3}$$

Where

i_d , i_q – stator current in d and q – axis,

L_d , L_q – inductances of d and q – axis,

ω_r – rotor mechanical speed,

p_n – number of pole pairs,

ψ_f – permanent magnet flux.

(2) Modeling in $\alpha-\beta$ stationary frame

It's complex to perform the mathematical modeling of IPMSM in stationary frame. Here give some equations as:

$$\begin{cases} \psi_\alpha = \int (u_\alpha - Ri_\alpha)\, dt \\ \psi_\beta = \int (u_\beta - Ri_\beta)\, dt \end{cases} \tag{1-4}$$

$$\psi_s = \sqrt{\psi_\alpha{}^2 + \psi_\beta{}^2} \tag{1-5}$$

$$T_e = \frac{2}{3} p_n \left(\psi_\alpha i_\beta - \psi_\beta i_\alpha \right) \tag{1-6}$$

Where

u_α, u_β – stator voltage of α and β – axis,

i_α, i_β – stator current of α and β – axis,

ψ_α, ψ_β – stator flux of α and β – axis,

R – stator resistance,

ψ_s – amplitude of stator flux.

The dynamic equation is given as:

$$J \frac{d}{dt} \omega_r = \left(T_e - T_L - B\omega_r \right) \tag{1-7}$$

Where

J – combined inertia of rotor and load

B – combined viscous friction of rotor and load

T_L – shaft mechanical (load) torque

1.3 Outline of the Thesis

The contents of the thesis are organized as follows:

In Chapter 2, the basic principle of the conventional DTC algorithm for IPMSM is introduced. Next, predictive calculator of voltage vector reference is presented and SVM technique is introduced briefly. Finally, the super twisting sliding mode control is introduced and STSM based DTC technique (STSM-DTC) is designed.

In Chapter 3, the currently popular artificial neural network control is presented briefly at first. Then, the proposed neural network based PI speed controller (NN-PI) for IPMSM is designed and presented in detail.

In Chapter 4, simulation works are designed and introduced in various scenarios. In this part, the comparisons between proposed control strategies and other methods are introduced to demonstrate the effectiveness and the improvement of proposed algorithm.

In Chapter 5, conclusions and future works are presented.

CHAPTER 2
DTC FOR PMSM

2.1 Review of Classical DTC

DTC is a significant new concept developed by ABB company. In this scheme, the motor parameters, namely electromagnetic torque and flux are being controlled directly. It doesn't need a modulator, as used in PWM drives, to control the frequency and voltage. The schematic of conventional DTC for PMSM is shown in Fig. 1. It includes two parts, namely speed control in the outer loop and torque control in the inner loop. It usually consists of a proportional-integral (PI) speed controller, two hysteresis controllers for electromagnetic torque and stator flux regulation, heuristic switching table for generating a variable switching frequency and electromagnetic torque and stator flux estimator. The principle of DTC for voltage source inverter (VSI)-fed PMSM is presented in the followings of this section.

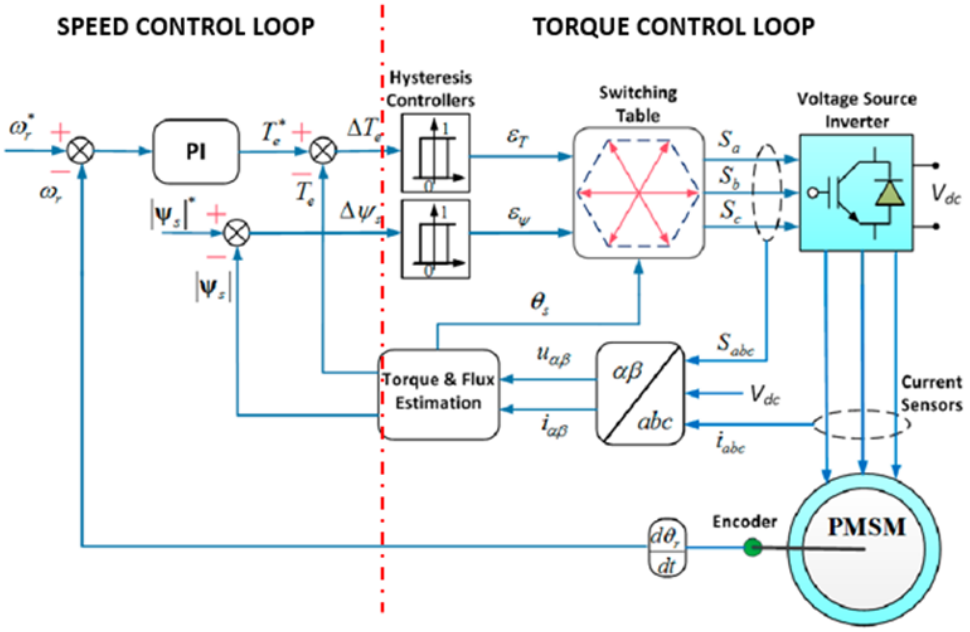


Fig. 1. Block diagram of the conventional DTC for PMSM.

Firstly, the principles of torque control will be presented. Fig. 2. shows some vector in different reference frames. The stator flux for a PMSM in $d-q$ rotating coordinate are given as:

$$\begin{cases} \psi_d = |\psi_s| \cos\delta \\ \psi_q = |\psi_s| \sin\delta \end{cases} \tag{2-1}$$

Where δ is torque angle between stator flux and permanent magnet flux.

According to the formula (1-2) and (2-1), the stator current of referencing to $d-q$ axis can be expressed as:

$$\begin{cases} i_d = \dfrac{\psi_d - \psi_f}{L_d} = \dfrac{|\psi_s| \cos\delta - \psi_f}{L_d} \\ i_q = \dfrac{\psi_q}{L_q} = \dfrac{|\psi_s| \sin\delta}{L_q} \end{cases} \tag{2-2}$$

Based on the equations (1-3) and (2-2), the torque can be described as the following equation [2]:

$$T_e = \frac{3}{2}\frac{p_n}{L_d}|\psi_s|\psi_f \sin\delta + \frac{3(L_d - L_q)}{4L_d L_q}|\psi_s|^2 \sin 2\delta \tag{2-3}$$

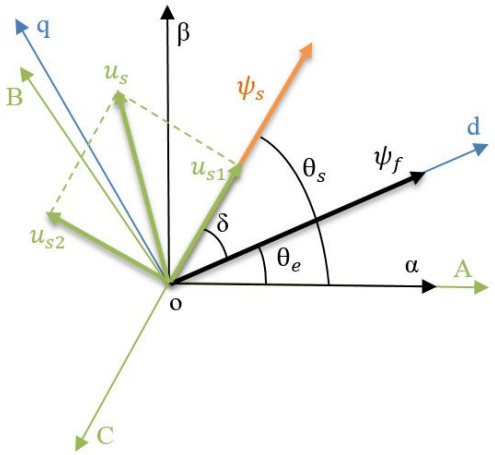


Fig. 2. Relationship between different quantities in different reference frames.

In a control period, rotor flux changes slowly due to its dependence on a relatively large rotor time constant; therefore, it can be assumed to be constant in short time. In DTC control scheme, the stator flux amplitude is also a constant value. From equation (2-3), it can be concluded torque can be changed if the instantaneous positions of the stator flux vector are changed quickly so that the torque angle δ is altered as quickly as possible [19]. This is the basis of DTC method. The stator flux vector instantaneous change can be achieved by switching on the appropriate stator voltage space vector of the VSI. The stator flux vector in stationary $A-B-C$ coordinate can be expressed as following:

$$\vec{\psi}_s = \int \left(\vec{u}_s - R_s \vec{i}_s\right) dt \tag{2-4}$$

Where $\vec{u}_s$ and $\vec{i}_s$ are stator voltage vector and current vector in stationary $A-B-C$ coordinate. If neglecting the stator resistance voltage dropping, in a short period of time the relationship between incremental $\Delta\vec{\psi}_s$ and $\vec{u}_s$ is proportional, given as:

$$\Delta\vec{\psi}_s = \vec{u}_s \Delta t \tag{2-5}$$

From (2-5), we can infer that changing rate of $\vec{\psi}_s$ is nearly same to that of $\vec{u}_s$. If define $\vec{\psi}_s = |\psi_s| e^{j\theta_s}$, where θ_s is the position angle of stator flux vector referenced to $A-$ axis, as shown in Fig. 2. Then the stator voltage vector can be described as:

$$\vec{u}_s = u_{s1} + u_{s2} = \frac{d|\psi_s|}{dt} e^{j\theta_s} + j\omega_s |\psi_s| \tag{2-6}$$

Where $\omega_s = \frac{d\theta_s}{dt}$ is rotating speed synchronous to $\vec{\psi}_s$. As shown in (2-6) and Fig. 2., $\vec{u}_s$ has two parts, namely u_{s1} and u_{s2}. u_{s1} coincides with $\vec{\psi}_s$; therefore, it can only change the amplitude of $\vec{\psi}_s$. How the stator flux is kept within a hysteresis band $|\Delta\psi_s|$ by selecting the voltage vectors is shown in Fig. 3. u_{s2} is perpendicular to $\vec{\psi}_s$; therefore, it can only control the rotating speed ω_s of $\vec{\psi}_s$, then finally control the electromagnetic torque. This the reason that DTC has the merit of fast dynamic control.

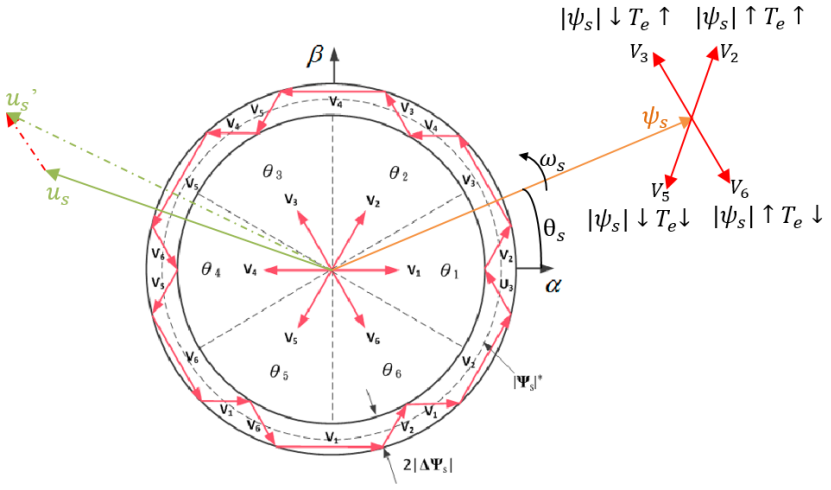


Fig. 3. Space voltage vectors and application to control stator flux space vector and torque.

As shown in Fig. 3., there are six sections with 60 degrees from θ_1 to θ_6 and only six active voltage vectors (two zero voltage vectors) V_i ($i = 1,2,3....8$), named the space vectors of inverter output phase voltages. It also illustrates that how the VSI output voltage vectors effect on the stator flux and torque,

where the stator flux vector $\vec{\psi}_s$ is located at sector θ_1 and rotate anticlockwise. The currently applied voltage vector is u_s. The selection process of voltage vectors is determined by inverter operation. If voltage vector $V_2(110)$ is selected, the torque and stator flux magnitude will be increased simultaneously. If voltage vector $V_5(001)$ is selected, the torque and stator flux magnitude will be decreased simultaneously. If applying $V_3(010)$, the torque will be increased but stator flux magnitude is decreased. Similarly, if applying $V_6(101)$, the torque will be reduced and stator flux magnitude is increased. This kind analysis can be extended similarly to the other sectors accordingly [2]. According to above analysis, the stator flux vector and torque can be controlled by using appropriate stator voltage space vectors, obtained from the VSI controlled by switching table, as shown in Fig. 3. Obviously, switchover of voltage vectors is operated step by step, and the stator flux vector is nearly the integration of voltage vector (2-4), thus the angle between stator flux and voltage vector varies step by step. However, the position of stator flux vector is continuously varying during the motor rotor rotates in practice. Hence, conventional DTC has the relatively large ripples in torque and flux.

The status of switches S_a, S_b and S_c of switching table decide the inverter output voltage space vectors. Fig. 4. shows the two level, three-phase VSI-fed PMSM system. The relationship between three-phase stator voltage vector $\vec{u}_s$ and switching states S_a, S_b and S_c can be described as equation (2-7). The optimum switching table of eight-voltage vector for controlling $\vec{\psi}_s$ and T_e shown in Table 1.

$$\vec{u}_s = \begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix} = \frac{U_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} S_a \\ S_b \\ Sc \end{bmatrix} \tag{2-7}$$

Where U_{dc} is dc-link voltage.

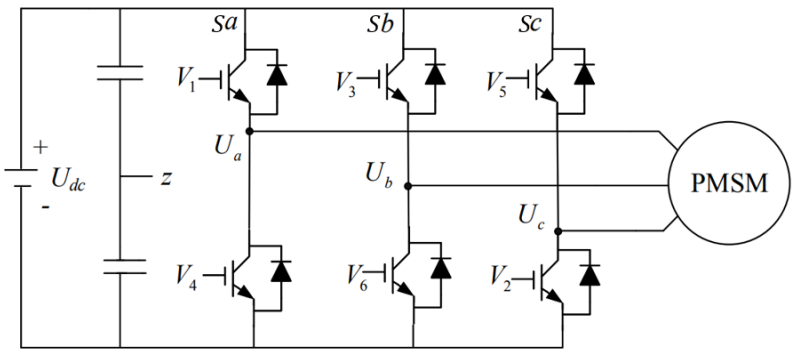


Fig. 4. Typical topology of three-phase VSI for PMSM.

The $\varepsilon\psi_s$, εT_e and θ_s are the inputs of switching table. $\varepsilon\psi_s$ and εT_e are generated by torque and flux hysteresis controllers respectively, which are described as following:

$$\varepsilon\psi_s = \begin{cases} 1 & if\ |\vec{\psi}_s| \le |\vec{\psi}_s^{\,*}| - |\Delta\psi_s| \\ 0 & if\ |\vec{\psi}_s| \ge |\vec{\psi}_s^{\,*}| + |\Delta\psi_s| \end{cases} \tag{2-8}$$

$$\varepsilon T_e = \begin{cases} 1 & if T_e \le T_e^{\,*} - |\Delta T_e| \\ 0 & if T_e = T_e^{\,*} \\ -1 & if T_e \ge T_e^{\,*} + |\Delta T_e| \end{cases} \tag{2-9}$$

Where $|\psi_s^{\,*}|$ and $|T_e^{\,*}|$ are reference stator flux magnitude and reference electromagnetic torque respectively. $\Delta\psi_s$ and ΔT_e are hysteresis band of stator flux magnitude and reference electromagnetic torque respectively.

Table 1. The optimum voltage vector look-up table [19]

$\varepsilon\psi_s$	εT_e	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	1	V_2	V_3	V_4	V_5	V_6	V_1
	0	V_7	V_8	V_7	V_8	V_7	V_8
	-1	V_6	V_1	V_2	V_3	V_4	V_5
0	1	V_3	V_4	V_5	V_6	V_1	V_2
	0	V_8	V_7	V_8	V_7	V_8	V_7
	-1	V_5	V_6	V_1	V_2	V_3	V_4

In the speed control loop, the reference torque is generated by the PI speed controller that regulate the speed error between reference speed and estimated speed.

2.2 Proposed STSM-DTC Technique Design

(1) Generation of Reference Voltage Vector

As introduced in section 2.1, the basic principle of DTC is to select the optimal voltage vectors, which makes the stator flux linkage space vector rotates to required position while keeping magnitude within limited and generate desired torque. However, the conventional DTC suffers the limited number of voltage vectors with changeless amplitude and fixed direction [2]. The SVM algorithm will be adopted to control the VSI generate more reference voltage vectors. It can regulate the stator flux and torque more precisely with fixed switching frequency when incorporating with SVM technique [10].

In stationary $\alpha-\beta$ reference frame, reference flux vector calculator (RFVC) [12] is employed to produce reference stator flux vector $\vec{\psi}_{ref}$, then used for generating desired voltage vector $\vec{u}_{ref}$, as shown in Fig. 5. At sample time k, estimated stator flux amplitude $\psi_s(k)$ and its position angle $\theta_s(k)$ can be calculated from flux estimator. Then at next time $k+1$,these values become $\psi_s(k+1)$ and $\theta_s(k+1)$. The incremental angle $\Delta\theta_s$ between $\psi_s(k)$ and $\psi_s(k+1)$ is generated by STSM torque controller. If define $\psi_s(k+1)$ as same to given stator flux amplitude ψ_s^*, and $\psi_s(k)=\psi_s$. Then $\psi_s(k+1)$ and $\psi_s(k)$ can be described as followings:

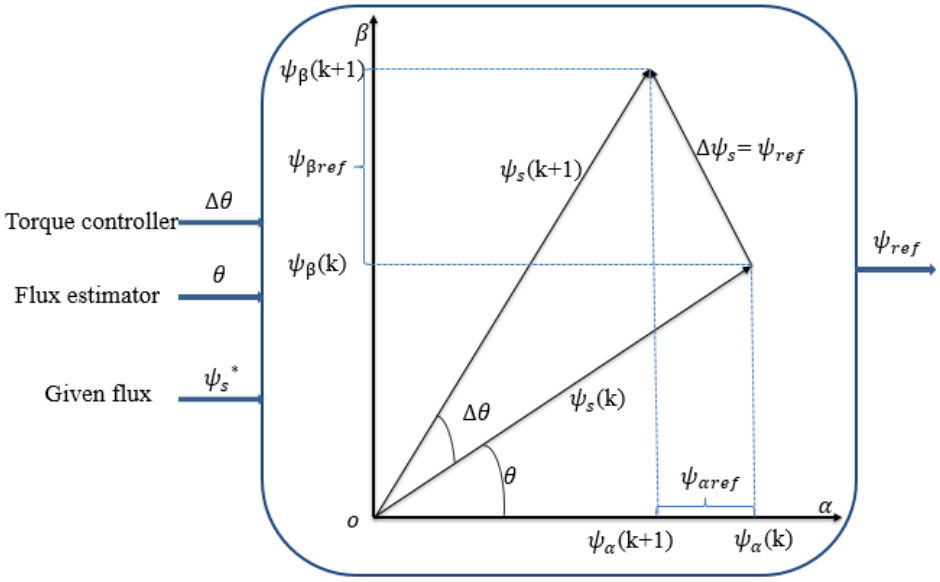


Fig. 5. Signal flow chart of generating reference flux vector [27].

$$\psi_s(k+1)=\begin{bmatrix}\psi_\alpha(k+1)\\ \psi_\beta(k+1)\end{bmatrix}=\begin{bmatrix}|\psi_s^*|\cos(\theta_s+\Delta\theta)\\ |\psi_s^*|\sin(\theta_s+\Delta\theta)\end{bmatrix} \tag{2-10}$$

$$\psi_s(k)=\begin{bmatrix}\psi_\alpha(k)\\ \psi_\beta(k)\end{bmatrix}=\begin{bmatrix}|\psi_s|\cos(\theta_s)\\ |\psi_s|\sin(\theta_s)\end{bmatrix} \tag{2-11}$$

Through the discretization of equation (2-4), the stator voltage vector can be transformed as following:

$$\vec{u}_s(k+1)=\frac{\vec{\psi}_s(k+1)-\vec{\psi}_s(k)}{T_s}+\vec{i}_sR \tag{2-12}$$

Where T_s is the sampling interval. $\vec{u}_s(k+1)$ is the desired voltage vector, namely reference stator voltage vectors. From equations (2-10), (2-11) and (2-12), we can get the reference voltage vector in $\alpha-\beta$ stationary reference frame as:

$$\begin{cases}u_{\alpha ref}=\dfrac{|\psi_s^*|\cos(\theta_s+\Delta\theta)-|\psi_s|\cos(\theta_s)}{T_s}+Ri_\alpha(k)\\ u_{\beta ref}=\dfrac{|\psi_s^*|\sin(\theta_s+\Delta\theta)-|\psi_s|\sin(\theta_s)}{T_s}+Ri_\beta(k)\end{cases} \tag{2-13}$$

By specifying a reference voltage vector, the SVM (space vector modulation) technique is used to determine action time and produce switching control signals to be applied to the inverter [10]. Thus, a voltage vector will be reconstructed by switching on two adjacent vectors for proper time. One example that the reference voltage vector locates at sector I is shown in Fig. 6., where the vector $u_s(k+1)$ can be obtained by the different switching on/off sequence of fundamental voltage vectors $V_1/V_2/V_0$ with time duration $T_1/T_2/T_0$. This is the basis of SVM, called as average equivalence principle, as described in equation (2-14). This proposed DTC with SVM technique still retains all the advantages of conventional DTC by replacing two hysteresis controllers with a STSM torque controller and predictive controller, the switching table with a SVM.

$$T_su_s=T_1V_1+T_2V_2+T_0V_0\cdots\cdots(T_1+T_2+T_0=T_s) \tag{2-14}$$

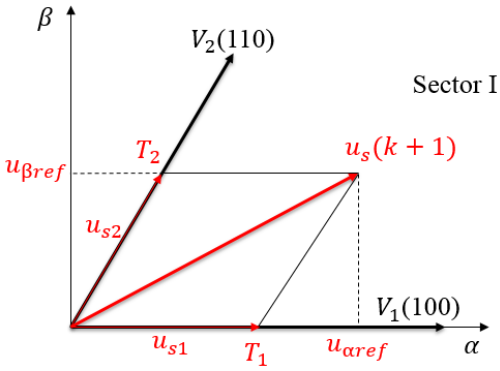


Fig. 6. Reference voltage vector as a combination of adjacent vectors at sector I.

(2) Design of Super-twisting SMC based Torque Controller

As known from the equation (2-3), the relation between the error of torque and the incremental of torque angle $\Delta\delta$ is nonlinear. In [12], the classical PI controller is applied to produce the torque angle increment required to minimize the error between reference torque and estimated torque. However, as mentioned before, the PI controller performs in nonlinear systems is not as good as that in linear systems.

In this paper, a novel super-twisting sliding mode (STSM) controller is designed and used to regulate the angle increment of $\Delta\theta$ which is linear to $\Delta\delta$. The super twisting is a modified second order sliding mode algorithm that can reduce the chattering effect while retaining the other properties of conventional SMC. Moreover, it does not need the information of any derivative of the sliding variable [9]. The general STSM control law contains two parts, one is a discontinuous function of the sliding variable, and the other is a continuous function of its derivative with respect to time, given as [25]:

$$\begin{cases} u_{ST} = -k_P \left| y \right|^r \operatorname{sgn}(y) + u_1 \\ \dfrac{du_1}{dt} = -k_I \operatorname{sgn}(y) \end{cases} \tag{2-15}$$

Where the k_P and k_I are positive gains used to adjust the STSM controller. The degree of nonlinearity of system can be adjusted by the positive coefficient $r(0 < r \le 0.5)$. The sufficient condition for a short time convergence to the sliding surface and stability are given by Levant in [26] as:

$$\begin{cases} k_I > \dfrac{A_M}{B_{\mathrm{m}}} \\ k_P \ge \dfrac{4A_M}{{B_{\mathrm{m}}}^2} \bullet \dfrac{B_M (k_I + A_M)}{B_m (k_I - A_M)} \end{cases} \tag{2-16}$$

Where $A_M \ge A$ and $B_M \ge B \ge B_m$,where A and B are defined by the second derivative of y :

$$\frac{d^2 y}{dt^2} = A(x,t) + B(x,t)\frac{du}{dt} \tag{2-17}$$

The first derivative of electromagnetic torque in equation (2-3) can be described as:

$$\frac{dT_e}{dt} = \frac{3p_n \left|\psi_s\right|}{4L_d L_q}\left[2\psi_f L_q \dot{\delta}\cos\delta + 2\left|\psi_s\right|(L_q - L_d)\dot{\delta}\cos(2\delta)\right] \tag{2-18}$$

It can be inferred that its second derivative contains the torque angle δ second derivative, that is torque angle incremental $\Delta\delta$ first derivative. Also the quantities A and B in (2-17) are both bounded, since the stator flux magnitude $\left|\psi_s\right|$ and inductions L_d , L_q are constants. As indicated in [12] that

$\Delta\theta = T_s p_n \omega_r + \Delta\delta$, where electrical angular speed ω_r is also limited. Therefore, STSM controller for angle incremental $\Delta\theta$ can be designed as:

$$\begin{cases} \Delta\theta = -k_P \left| s_{Te} \right|^{\frac{1}{2}} \tanh(a s_{Te}) + u_1 \\ \dfrac{du_1}{dt} = -k_I \tanh(a s_{Te}) \end{cases} \tag{2-19}$$

Where hyperbolic tangent function $\tanh(ax) = \dfrac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}} (a > 0)$ is adopted to replace the switching function $\mathrm{sgn}(x)$ which has strong discontinuity causing chattering. The applied $\tanh(ax)$ is continuous function with smoother variation of curve slope rate, which can eliminate the chattering. The coefficient a determines slope rate. The sliding variable is the torque error $s_{Te} = T_e^* - T_e$ between reference torque and actual torque. The gains k_P and k_I should fulfill the stability condition in equation (2-16). The diagram of proposed STSM torque controller is shown in Fig. 7. This modified DTC scheme is named as STSM-DTC in this paper.

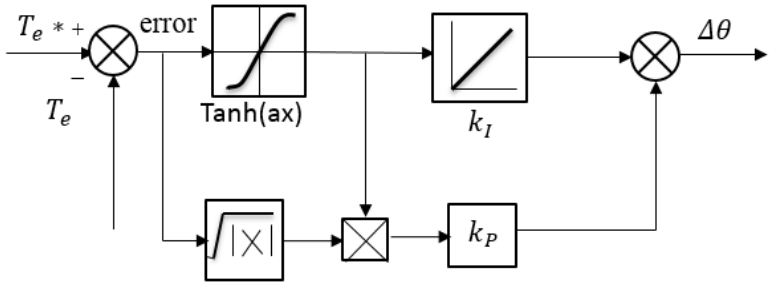


Fig. 7. The schematic of proposed STSM torque controller.

CHAPTER 3
SPEED CONTROLLER BASED ON ANN

3.1 ANN Control Technique

Natural neurons, the simple computational units of brain, which are interconnected in complex manner. Many interconnected neurons constitute neural network that can solve complex problems, replicate responses and generalize signal. Inspired by this, artificial neuron and ANN are created.

Fig. 8. shows the natural neural and artificial neuron. The artificial neuron basically consists of three parts. At the entrance, *inputs* are multiplied by individual *weights* that means strength of the respective signals. and then. In the middle, all previously weighted inputs and bias will be summed. At the exit of artificial neuron, the summed values will be computed by a mathematical function, namely activation function of the neuron [20].

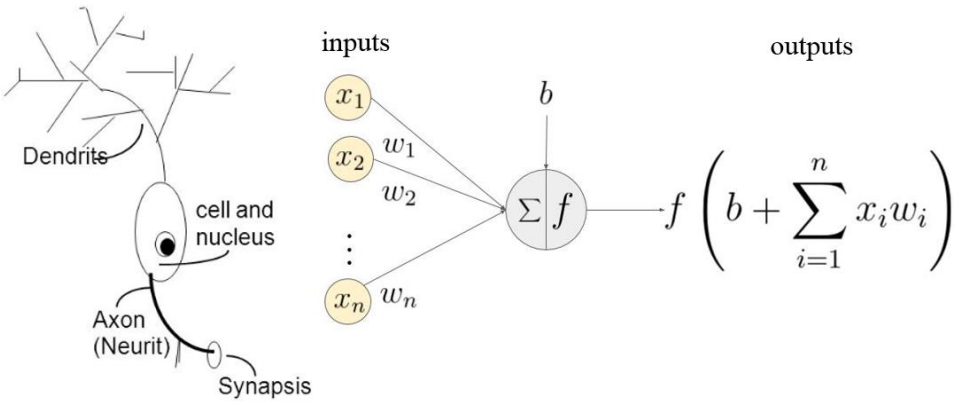


Fig. 8. Biological neuron (left) and artificial neuron (right).

ANN can be built to solve complex problems when combining two or more artificial neurons. According to the way of artificial neurons' interconnection, ANNs are classified into two basic types, Feed-forward Neural Network (FNN) and Recurrent Neural Network (RNN). The simple examples of three-layered FNN and RNN are shown in Fig. 9. Multi-Layer Perceptron Neural Network (MLPNN)is one typical FNN, which can be used to represent any nonlinear mapping to an arbitrary degree of

accuracy. In this paper, the MLPNN is adopted in motor speed control.

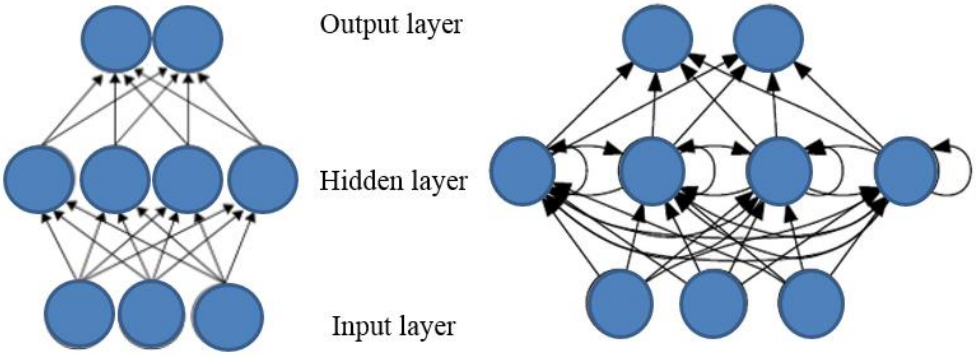


Fig. 9. The topology schematic of three-layered FNN (left) and RNN (right).

As known that biological neural networks can tell how to respond to the given inputs from the environment properly after learning. Therefore, to get the expected outputs from ANNs, we need to teach the ANNs how to learn from processing experience. There are three major learning paradigms, namely supervised learning, unsupervised learning and reinforcement learning. All these learning method target to set the values of weight and biases on basis of learning data to minimize the chosen cost function [21]. Supervised learning is a widely used machine learning technique. Its task is to set parameters of an artificial neural network based on the training data which consists of inputs and desired outputs.

The Back-Propagation (BP) algorithm is the most common learning algorithm of supervised learning technique, typically using gradient descent method. The idea of BP algorithm is to reduce the error between actual outputs and expected results, until the ANN learns the training data [20]. The structure of a three-layer BP neural network is shown in Fig. 10.

Notation states:

- n_l: number of neurons in l^{th} layer;
- $f(\bullet)$: activation of neuron;
- $\mathbf{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$: weights vector from $(l-1)^{th}$ layer to l^{th} layer;
- $w_{ij}^{(l)}$: the element of weights vector $W^{(l)}$, representing the weight between j^{th} neuron in

$(l-1)^{th}$ layer and i^{th} neuron in l^{th} layer;

- $\mathbf{b}^{(l)} = \left(b_1^{(l)}, b_2^{(l)}, \cdots, b_{n_{(l)}}^{(l)}\right)^{\mathrm{T}} \in \mathbb{R}^{n_{(l)}}$: the bias between $(l-1)^{th}$ layer and l^{th} layer;
- $\mathbf{z}^{(l)} = \left(z_1^{(l)}, z_2^{(l)}, \cdots, z_{n_{(l)}}^{(l)}\right)^{\mathrm{T}} \in \mathbb{R}^{n_{(l)}}$: states of neurons in l^{th} layer;
- $\mathbf{a}^{(l)} = \left(a_1^{(l)}, a_2^{(l)}, \cdots, a_{n_l}^{(l)}\right)^{\mathrm{T}} \in \mathbb{R}^{n_l}$: outputs of neurons in l^{th} layer.

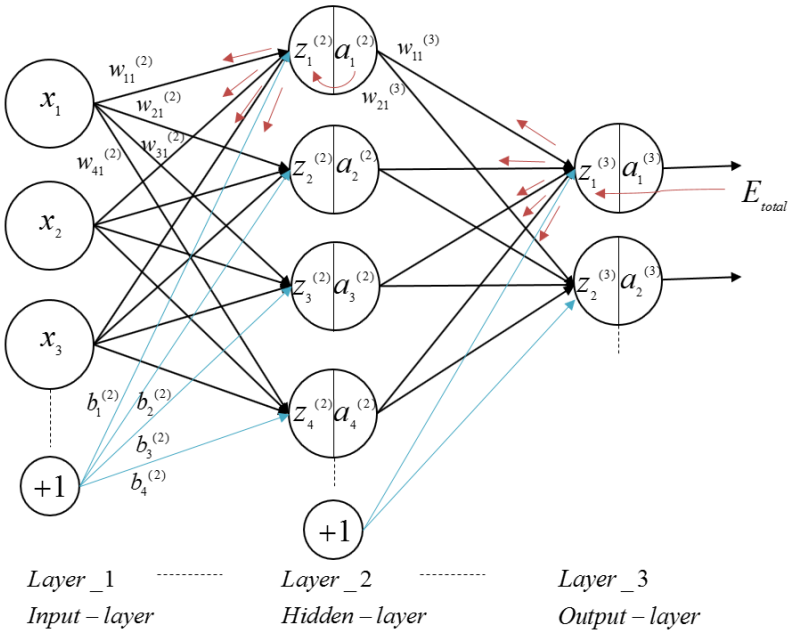


Fig.10. The structure of BP neural network.

If the BP neural network is a three-layered model with j inputs, i hidden nodes and k outputs. Then the work process of BP algorithm is given as following [22]:

Step 1: Initialize weight to small random values.
Step 2: While stopping condition is false, do steps 3-10.
Step 3: For each training pair do steps 4-9.
Feedforward pass

Step 4: the state vector and activation vector in l^{th} $(2 \le l \le L)$ layer is expressed as:

$$\begin{aligned} \mathbf{z}^{(l)} &= \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \\ \mathbf{a}^{(l)} &= f\left(\mathbf{z}^{(l)}\right) \end{aligned} \tag{3-1}$$

For a L^{th} layer perceptron, the final output of network is $\mathbf{a}^{(L)}$. The feedforward process of information is described as:

$$\mathbf{x} = \mathbf{a}^{(1)} \to \mathbf{z}^{(2)} \to \cdots \mathbf{a}^{(L-1)} \to \mathbf{z}^{(L)} \to \mathbf{a}^{(L)} \to \mathbf{y} \tag{3-2}$$

Backward pass

Step 5: Set the training data as $\left\{\left(\mathbf{x}^{(1)},\mathbf{y}^{(1)}\right),\left(\mathbf{x}^{(2)},\mathbf{y}^{(2)}\right),\cdots\left(\mathbf{x}^{(i)},\mathbf{y}^{(i)}\right),\cdots\left(\mathbf{x}^{(N)},\mathbf{y}^{(N)}\right)\right\}$ with N data sets, and $\mathbf{y}^{(i)}=\left(y_1^{(i)},\cdots,y_{nL}{}^{(i)}\right)^{\mathrm{T}}$. For one training data $\left(\mathbf{x}^{(i)},\mathbf{y}^{(i)}\right)$, the cost function is as following:

$$E_{(i)}=\frac{1}{2}\left\|\mathbf{y}^{(i)}-\mathbf{o}^{(i)}\right\|=\frac{1}{2}\sum_{k=1}^{nL}\left(y_k^{(i)}-o_k^{(i)}\right)^2 \tag{3-3}$$

Where $\mathbf{y}^{(i)}$ is the expected output vector, $\mathbf{o}^{(i)}$ is the actual output vector according to the input of $\mathbf{x}^{(i)}$. If expand the equation (3-3) to hidden layers and finally input layer, we will find that the cost function is only related to weights vector $\mathbf{W}^{(l)}$ and bias vector $\mathbf{b}^{(l)}$, which means that adjusting the values of weights and bias can reduce or enlarge the error. Obviously, the totally averaged cost of all training data is:

$$E_{total}=\frac{1}{N}\sum_{i=1}^{N}E_{(i)} \tag{3-4}$$

Step 6: Update the weights and bias of l^{th} layer from time t to $t+1$ by using gradient descent method. The iteration updating equations of parameters are given as:

$$\begin{aligned}\mathbf{W}_{(t+1)}^{(l)}&=\mathbf{W}_{(t)}^{(l)}+\Delta\mathbf{W}_{(t+1)}^{(l)}\\&=\mathbf{W}_{(t)}^{(l)}-\frac{\eta}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)(t)}}{\partial\mathbf{W}_{(t)}^{(l)}}\end{aligned} \tag{3-5}$$

$$\begin{aligned}\mathbf{b}_{(t+1)}^{(l)}&=\mathbf{b}_{(t)}^{(l)}+\Delta\mathbf{b}_{(t+1)}^{(l)}\\&=\mathbf{b}_{(t)}^{(l)}-\frac{\eta}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)(t)}}{\partial\mathbf{b}_{(t)}^{(l)}}\end{aligned} \tag{3-6}$$

Where η is learning rate.

3.2 Proposed NN-PI Speed controller

Inspired by the amazing function of neurons, the artificial neural network (ANN) has been developed very fast to solve big scale and complex problems recently. Numerous works [16], [23-24] have demonstrated that ANN is suitable for dynamic and nonlinear control systems due to the capability of processing, analyzing and learning. The speed controller of DTC structure is a nonlinear system, which has disturbances of load inertia and motor parameters variation against temperature [17]. This makes it difficult for PI speed controller to perform sufficiently well for this DTC control system. This paper proposes an adaptive speed controller based on back propagation (BP) neural network, which is verified by simulation with high performances.

The PI control theory has been one of the most widely used methods in industrial application. In a continuous-signal system, the PI equation is expressed in (3-7). When the sampling period is very small, a discrete-time PI controller can be obtained as (3-8) by replacing integrate with summation and replacing derivative with difference quotient of (3-7).

$$u(t) = k_p(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau) \tag{3-7}$$

$$u(k) = k_p(e(k) + \frac{T_s}{T_i}\sum_{j=0}^{k} e(j)) \tag{3-8}$$

Where $u(t)$ and $u(k)$ are controller outputs in the inner training loop at continuous-time t and discrete-time k, respectively; $e(t)$ and $e(k)$ represent tracking errors at continuous-time t and discrete-time k, respectively; k_p is proportional gain; T_i is integral time constant; T_s is the sampling period. However, the calculating of integral part in equation (3-8) needs to accumulate all errors in the past, which takes huge storage and time. For convenience program coding work, equation (3-8) can be described as incremental form as following [24]:

$$\begin{aligned} u(k) &= u(k-1) + \Delta u(k) \\ &= u(k-1) + k_p\left(e(k) - e(k-1)\right) + k_i e(k) \end{aligned} \tag{3-9}$$

Where k_i is the integral gain. k_p and k_i are auto-tuning parameters depending on the system states. The equation (3-9) can be described as:

$$u(k) = f[u(k-1), k_p, k_i, e(k), e(k-1)] \tag{3-10}$$

Where $f(\bullet)$ is a kind of nonlinear function related to $u(k-1), y(k), k_p, k_i$. BP neural network can find the best control rule of $f(\bullet)$ through training and learning.

The NN-PI speed controller using error backpropagation (BP) algorithm consists of two parts. One is classical PI controller which can regulate motor speed in the closed loop. The other is BP neural network which can tune the parameters of k_p and k_i online to achieve the optimal performance [28]. The schematic structure of NN-PI speed controller is shown as in Fig. 11.

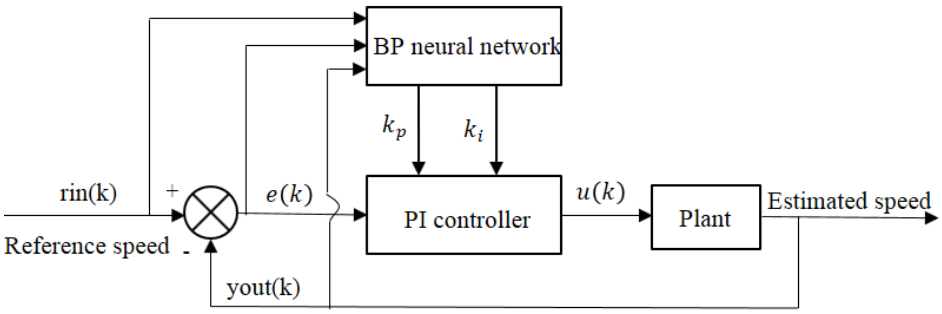


Fig. 11. The structure of NN-PI speed controller based on BP algorithm.

This NN-PI speed controller adopts a three-layers BP neural network with 3 input-neurons in input layer, 5 hidden-neurons in hidden layer and 2 output-neurons in output layer, as shown in Fig. 12. The inputs are set as reference speed, actual speed and error between them. The k_p and k_i are non-negative outputs of network. Sigmoid function is selected as the activation function of output layer.

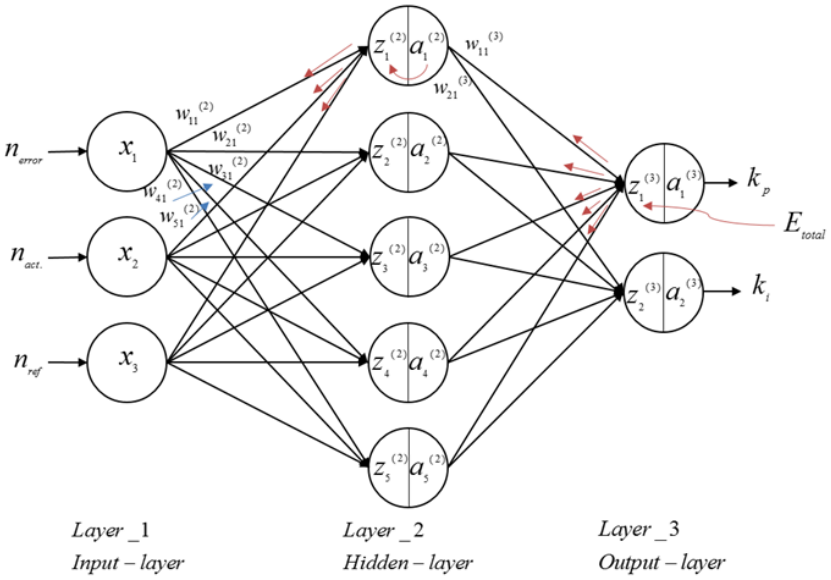


Fig. 12. The BP neural network for proposed speed controller.

Based on the section 3.1 and Fig. 12., the equations for proposed NN-PI speed controller are given as:

$$a_j^{(1)}(k) = x_j(k) \qquad (j = 1,2,3) \tag{3-11}$$

$$\begin{aligned} z_i^{(2)}(k) &= \sum_{j=1}^{3} w_{ij}{}^{(2)} a_j{}^{(1)}(k) \\ a_i{}^{(2)}(k) &= f(z_i^{(2)}(k)) \qquad (i = 1,2,3,4,5) \end{aligned} \tag{3-12}$$

$$\begin{aligned} z_k^{(3)}(k) &= \sum_{i=1}^{2} w_{ki}^{(3)} a_i^{(2)}(k) \\ a_k^{(3)}(k) &= g(z_k^{(3)}(k)) \qquad (k = 1,2) \end{aligned} \tag{3-13}$$

$$\begin{aligned} f(x) &= \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ g(x) &= \frac{1}{2}\left(1 + \tanh\left(x\right)\right) = \frac{e^x}{e^x + e^{-x}} \end{aligned} \tag{3-14}$$

Where $z_n^{(l)}$ and $a_n^{(l)}$ present the summed input values and output of n^{th} neuron in l^{th} layer, respectively. $w_{nm}^{(l)}$ denotes the connection weight between m^{th} neuron in $(l-1)^{th}$ layer and n^{th} neuron in l^{th} layer. $f(x)$ and $g(x)$ are activation function for hidden and output layer, respectively. The backpropagation pass is described as followings.

$$\Delta w_{ki}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{ki}^{(3)}} + \alpha \Delta w_{ki}^{(3)}(k-1) \tag{3-15}$$

Where $\Delta w_{ki}^{(3)}(k)$ is weight incremental quantity of output layer with an additional momentum for fast convergence. α is momentum coefficient. $E(k)$ is cost function. According to the chain rule of derivative, the partial derivative of error to weight can be expressed as:

$$\frac{\partial E(k)}{\partial w_{ki}^{(3)}} = \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial a_k^{(3)}(k)} \cdot \frac{\partial a_k^{(3)}(k)}{\partial z_k^{(3)}(k)} \cdot \frac{\partial z_k^{(3)}(k)}{\partial w_{ki}^{(3)}} \tag{3-16}$$

Wherein $y(k)$ stands for the actual outputs; $\frac{\partial y(k)}{\partial u(k)}$ is unknown as the model is unknown. But the variation amounts of $y(k)$ and $u(k)$ can be obtained. Therefore, the $\mathrm{sgn}(\bullet)$, that determines the direction of weights varying, can be employed to get the approximate value. The caused uncertainty can be compensated by adjusting learning rate η. According to $a_1^{(3)} = k_p$, $a_2^{(3)} = k_i$ and equation (3-9), we get:

$$\left.\begin{aligned}\frac{\partial u(k)}{\partial a_1^{(3)}(k)} &= e(k)-e(k-1)\\ \frac{\partial u(k)}{\partial a_2^{(3)}(k)} &= e(k)\end{aligned}\right\} \tag{3-17}$$

The equation (3-15) can be rewritten as:

$$\Delta w_{ki}^{(3)} = \eta e(k)\operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right)\frac{\partial u(k)}{\partial a_k^{(3)}(k)} g'(z_k^{(3)}(k)) a_i^{(2)}(k) + \alpha \Delta w_{ki}^{(3)}(k-1) \tag{3-18}$$

If define: $\delta_k^{(3)} = e(k)\operatorname{sgn}\left(\frac{y(k)-y(k-1)}{u(k)-u(k-1)}\right)\frac{\partial u(k)}{\partial a_k^{(3)}(k)} g'(z_k^{(3)}(k))$, Then:

$$\Delta w_{ki}^{(3)}(k) = \eta \delta_k^{(3)} a_k^{(2)}(k) + \alpha \Delta w_{ki}^{(3)}(k-1) \tag{3-19}$$

Similarly, the weight incremental quantity of hidden layer is given as:

$$\Delta w_{ij}^{(2)}(k) = \eta \delta_i^{(2)} a_j^{(1)}(k) + \alpha \Delta w_{ij}^{(2)}(k-1) \tag{3-20}$$

Where $\delta_i^{(2)} = f'(z_i^{(2)}(k))\sum_{k=1}^{2}\delta_k^{(3)} w_{ki}^{(3)}(k)$. The process of BP neural network is shown in Fig. 13.

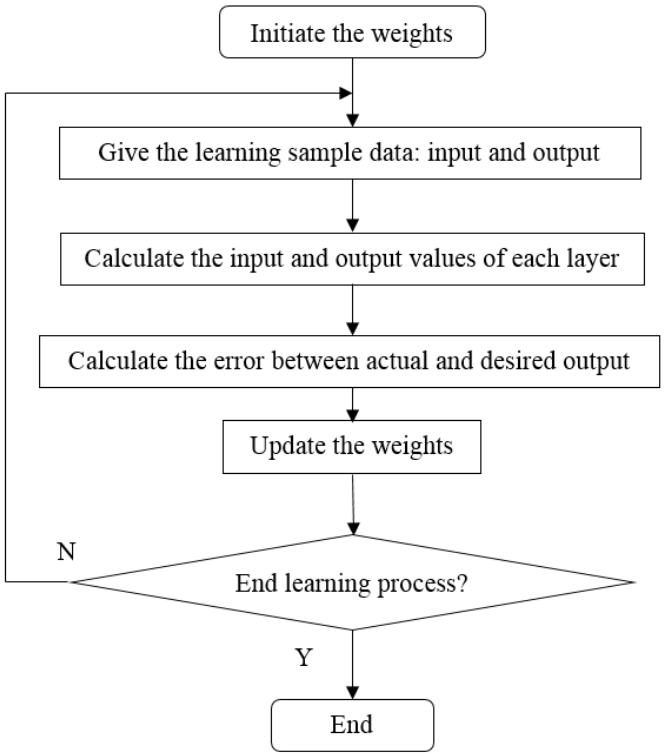


Fig. 13. The process of BP neural network.

CHAPTER 4
SIMULATION RESULTS

4.1 Simulation verification of STSM-DTC method

In this thesis, a novel super twisting sliding mode torque controller based predictive calculator for generating reference voltage vector (STSM-DTC) is proposed. To test the effectiveness of the proposed method, the evaluation and contrasted works to conventional DTC and the PI-DTC presented in previous research [27] are tested by simulation on Matlab/SIMULINK. The Simulink block of modified DTC based on STSM control for IPMSM is shown in Fig. 14. To compare the performances of STSM-DTC, PI-DTC and conventional DTC, all motor parameters, simulation settings and conditions are adjusted to the same. The IPMSM specification and system parameters are listed in Table 2.

Torque and flux ripples are taken as the most important measuring standard. In this paper, the performances of torque and flux are evaluated comprehensively by range (peak-to-peak) and ripple (standard deviation) shown in formula (4-1). The *Rip*. can be used to reflect the dispersion degree of the torque and flux samples.

$$Rip. = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_{(i)} - X_{mean})^2} \tag{4-1}$$

Where N is the number of samples, and variable X stands for torque or stator flux.

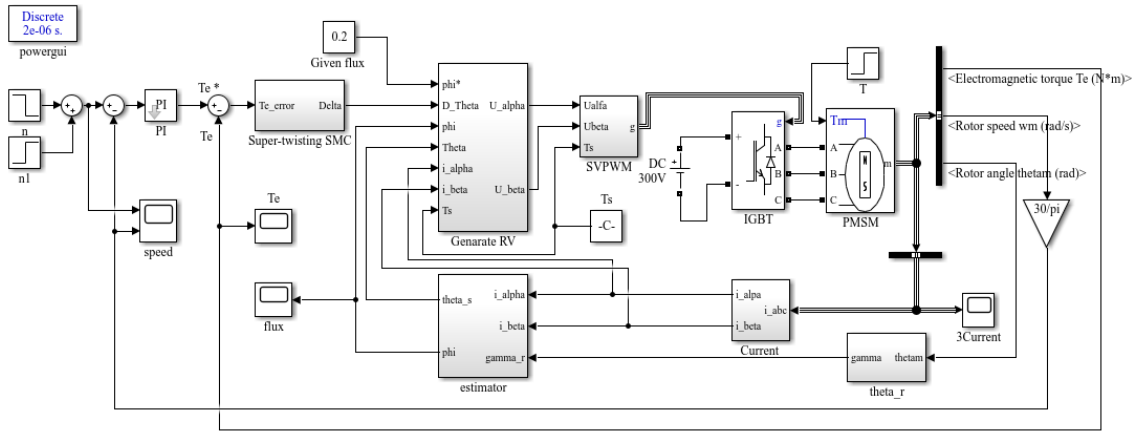


Fig. 14. Simulation block of STSM-DTC for IPMSM system.

Table 2. System parameters

Parameter	**Values**	**Unite**
DC-link voltage	300	V
Rated power	750	W
Rated speed	3000	rpm
Sampling time	2	μs
Number of pole pairs	4	-
Stator resistance	1.2	Ω
Magnet flux linkage	0.0686	Wb
d-axis inductance	5.5	mH
q-axis inductance	6.5	mH
Moment inertia	0.0004	$kg \cdot m^2$
Viscous friction	0.0001	$N \cdot m \cdot s$

(1) Select the optimal parameters

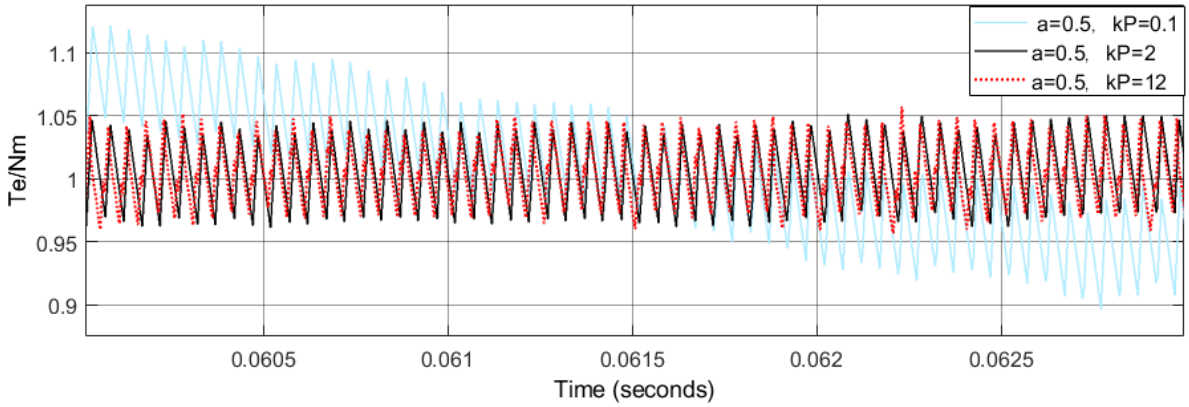
Before comparing with the other two methods, the optimal parameters of STSM torque controller should be selected through massive simulation works. The simulation settings and conditions are kept same with $600\, m/s$ as reference speed, $1\, Nm$ as applied load torque and $0.2\, Wb$ as reference stator flux magnitude. As found that the parameter of k_I has much smaller influence compared to the parameters of k_P & a . Therefore, the k_I is set as a constant 10. Parts of the selection process of k_P and a are shown in the following.

Set $a = 0.5$ and only change the value of k_P, the specific statistics of torque and flux in different cases according to the variations of k_P are shown in Table 3 and the partial results in steady state are shown detailly in Fig. 15. Roughly to see that increasing the value of k_P can influence the response of torque and performance of stator flux at different rate. clearly, continuously increasing the value of k_P (when $k_P \geq 2 \sim 5$) will degenerate the response of stator flux. For DTC scheme, torque-ripple reduction is more important.

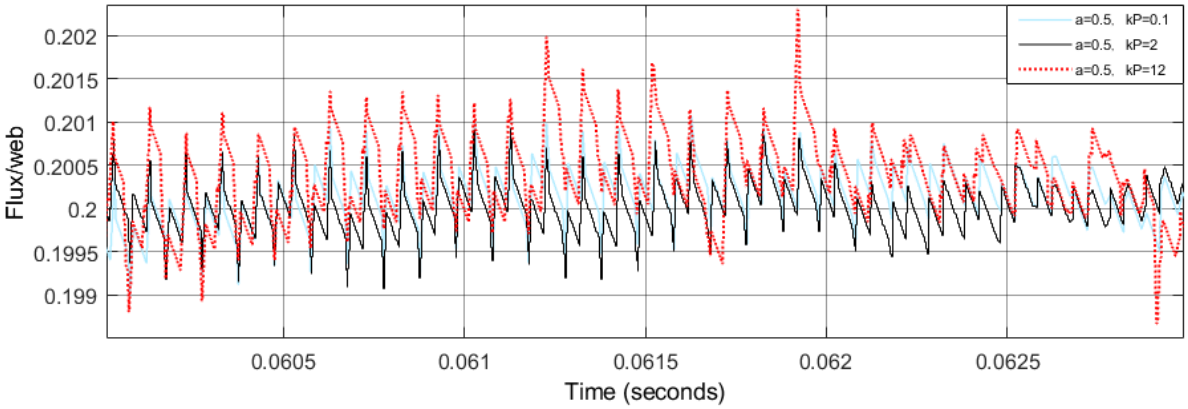
Table 3. The responses of torque and flux according to the variations of k_P

Results-		Kp=0.1	Kp=0.5	Kp=0.9	Kp=2	Kp=5	Kp=12
T_e	Range	2.49e-01	1.20e-01	1.25e-01	9.94e-02	9.80e-02	1.02e-01
	Rip.	5.06e-02	2.34e-02	2.36e-02	2.28e-02	2.27e-02	2.13e-02
ψ_s	Range	2.41e-03	2.32e-03	2.38e-03	2.22e-03	2.31e-03	4.44e-03
	Rip.	3.29e-04	3.03e-04	2.92e-04	2.79e-04	3.21e-04	5.57e-04

The trade-off strategy in parameter selection is considered. It can be inferred that choosing k_p between $2 \sim 5$ may get the optimal overall results. Finally, k_p is chosen as 3.



(a)



(b)

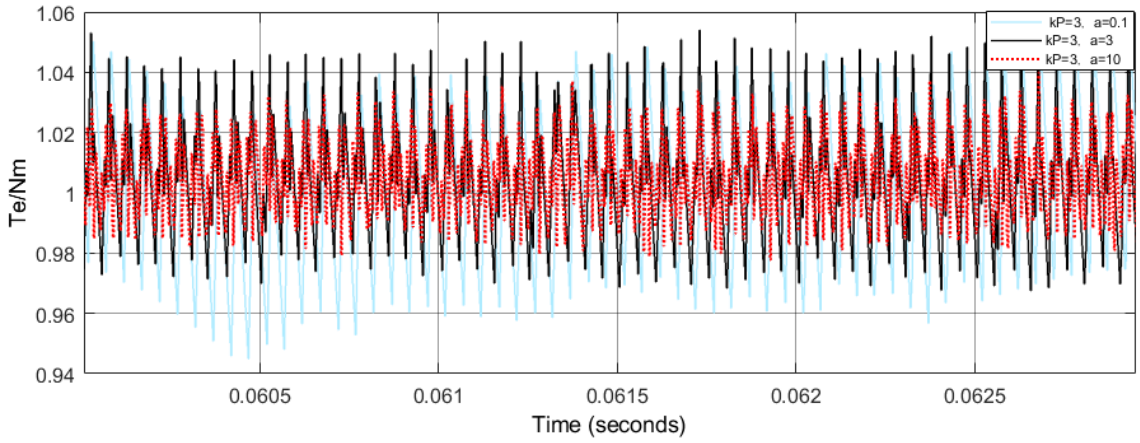
Fig. 15. Responses of torque (a) and flux (b)in steady state depending on variations of k_P.

Similarly, set $k_P = 3$ and only change a, the specific statistics of torque and flux in different cases according to a are shown in Table 4 and the partial results in steady state are shown detailly in Fig. 16.

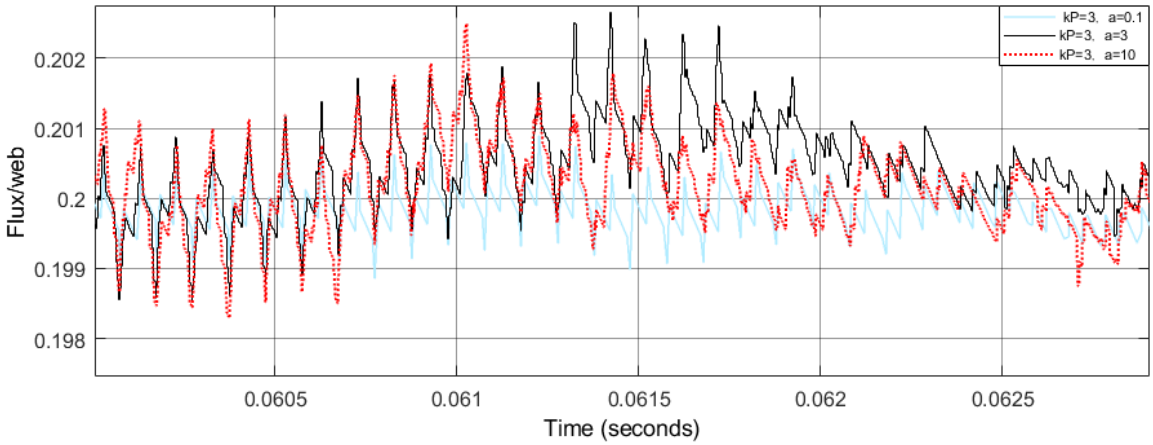
Table 4. The responses of torque and flux according to the variations of a

Results-		a=0.1	a=0.5	a=0.9	a=3	a=6	a=10
T_e	Range	1.26e-01	9.74e-02	1.01e-01	9.29e-02	7.23e-02	6.39e-02
	Rip.	2.43e-02	2.23e-02	2.23e-02	1.83e-02	1.34e-02	1.20e-02
ψ_s	Range	2.43e-03	2.39e-03	2.49e-03	4.76e-03	5.10e-03	4.69e-03
	Rip.	2.93e-04	3.11e-04	3.44e-04	6.90e-04	7.56e-04	7.03e-04

From the statistics in Table 3, 4 and Fig. 15, 16., it can be inferred that the system can obtain relatively good performances of torque and flux when choosing $k_P = 3$ and $a = 0.9$.



(a)



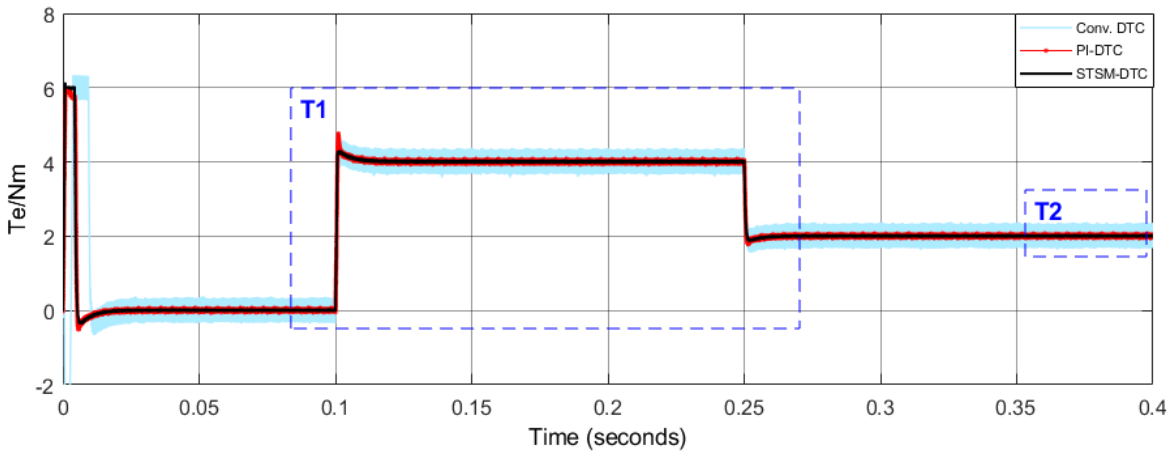
(b)

Fig. 16. Responses of torque (a) and flux(b) in steady state depending on variations of a.

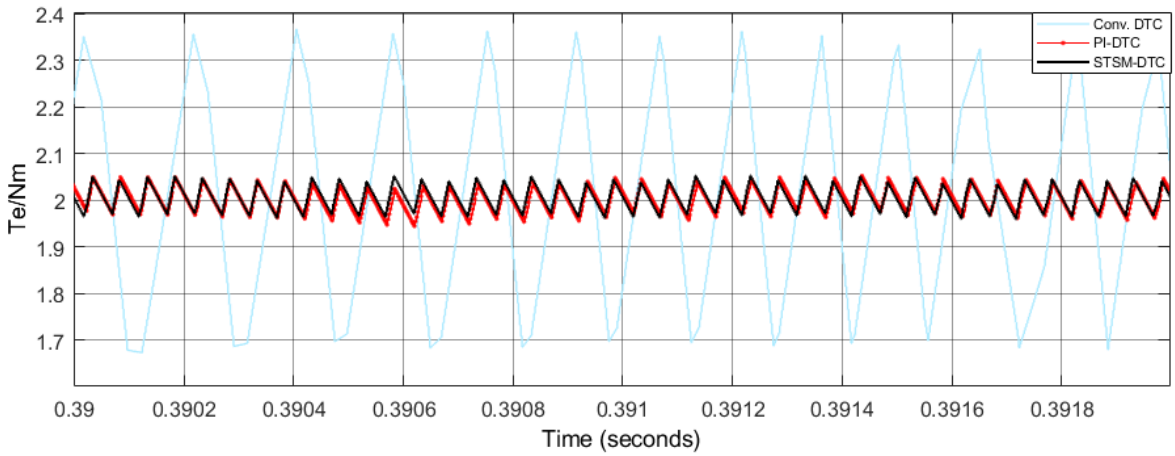
(2) comparison works

For validating the dynamic and steady-state responses of torque control of proposed STSM-DTC method for IPMSM. The three methods are compared with same operating situation, setting reference constant speed of 600 *rpm* , reference flux at 0.20 *Wb* and the torque command increased from 0 *Nm* to 4 *Nm* at 0.1 *s* then decreased to 2 *Nm* at 0.25 *s* .

The simulation results of torque and partial details for these three methods are shown in Fig. 17. Meanwhile, the statistical data of ZOOM -T1 (dynamic response) and ZOOM-T2 (steady-state response) are listed in Table 5. Obviously, in contrast to conventional DTC and PI-DTC methods, the proposed STSM-DTC technique has the best performances in steady torque response with smallest torque ripple and overshoot (rising edge), as well as fast torque response.



(a)



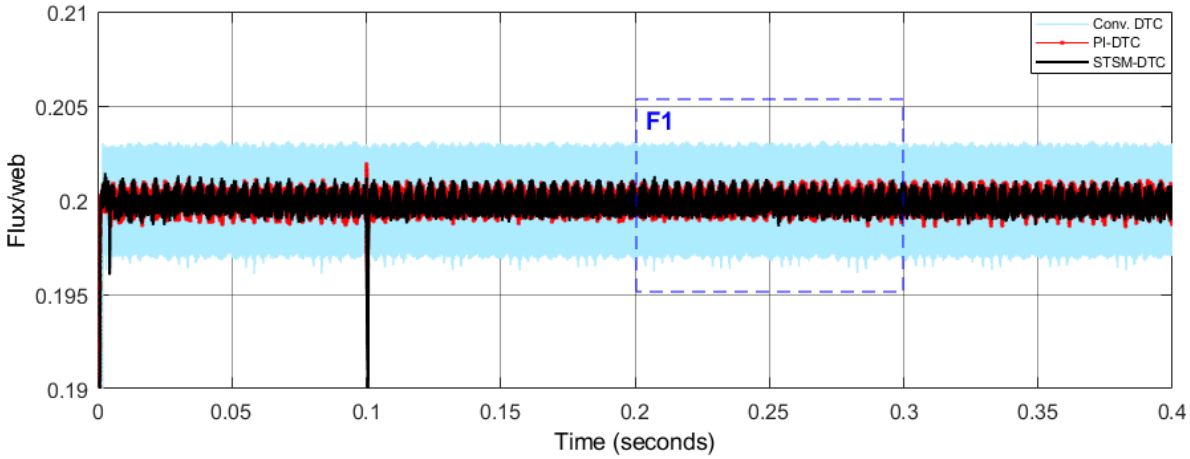
(b)

Fig. 17. Results of torque performances(a) and partial details of ZOOM-T2(b).

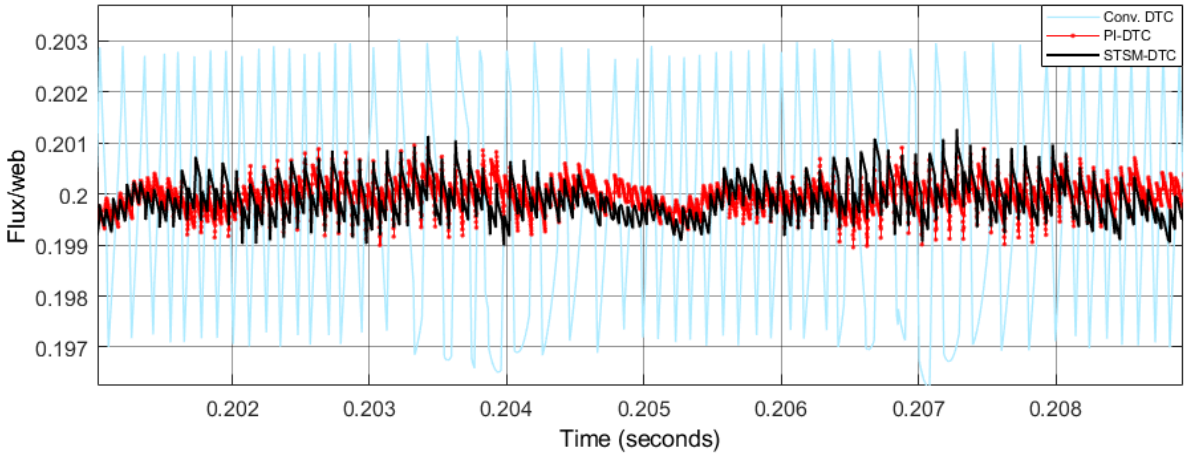
Table 5. The statistics of torque response in ZOOM-T1and ZOOM-T2

Result-		Conv. DTC	PI-DTC	STSM-DTC	Unit
T1	Rising time+	0.38	0.25	0.24	*ms*
	Overshoot+	40.42	36.90	15.86	%
	Falling time-	0.86	0.54	0.69	*ms*
T2	Range	7.52e-01	1.38e-01	1.02e-01	*Nm*
	Rip.	2.00e-01	2.51e-02	2.35e-02	*Nm*

Similarly, the simulation results of flux and partial details are shown in Fig. 18. Meanwhile, the statistical data of ZOOM -F1 is listed in Table 6. The proposed method can obtain much better results compared to conventional DTC, but the ripple is slightly bigger than PI-DTC method.



(a)



(b)

Fig. 18. Results of flux performances (a) and partial details of ZOOM-F1(b).

Table 6. The statistics of flux response in ZOOM-F1

Result-		Conv. DTC	PI-DTC	STSM-DTC	Unit
F1	Range	6.91e-03	2.09e-03	2.31e-03	*Wb*
	Rip.	1.90e-03	2.91e-04	3.84e-04	*Wb*

Speed performances of three methods are shown in Fig. 19. This figure depicts the waveforms of measured speed. From the ZOOM-S1 and ZOOM-S2, we can see the proposed STSM-DTC method can also improve the speed performance with fast speed response, relatively small undershoot against load torque variation and the smallest speed ripple compared to conventional DTC and PI-DTC.

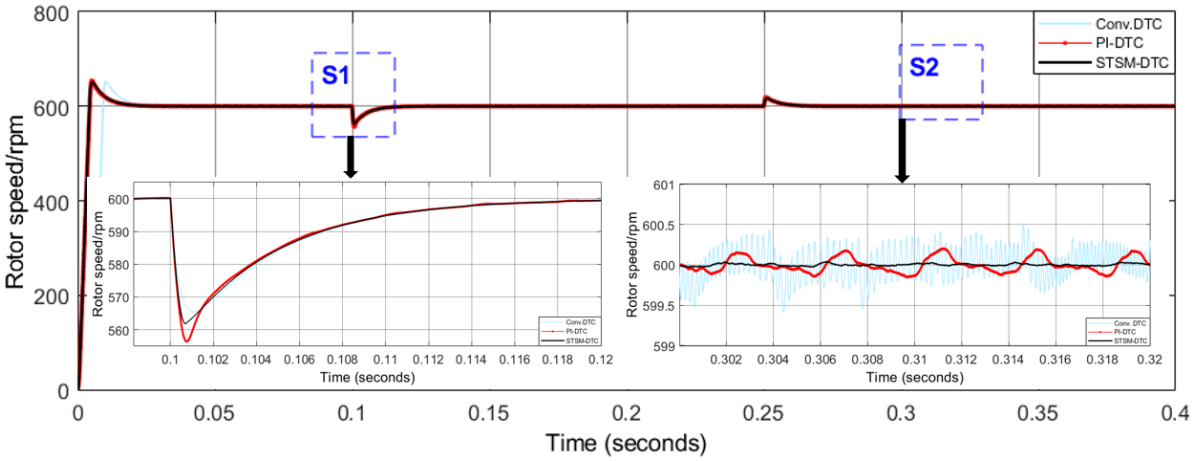


Fig. 19. Results of rotor speed performances.

As the IPMSM is highly nonlinear and coupled motor, the issue of power quality is also important in motor control, which is another aspect of research area of IPMSM drive. The Powergui block in Simulink can provide FFT (Fast Fourier Transform) analysis. The stator current is shown in Fig. 20., and FFT analysis and spectrum of THD (total harmonic distortion) is shown in Fig. 21.

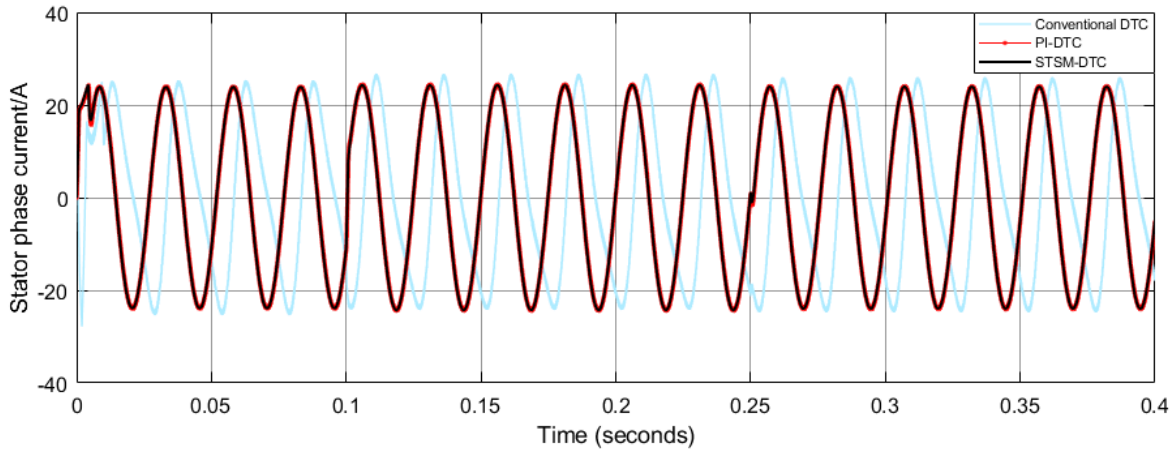


Fig. 20. Stator phase current i_{sa} (A) of three control schemes.

It’s clear that the proposed STSM-DTC technique can obtain good sinusoid waveform of stator current and relatively very low THD level.

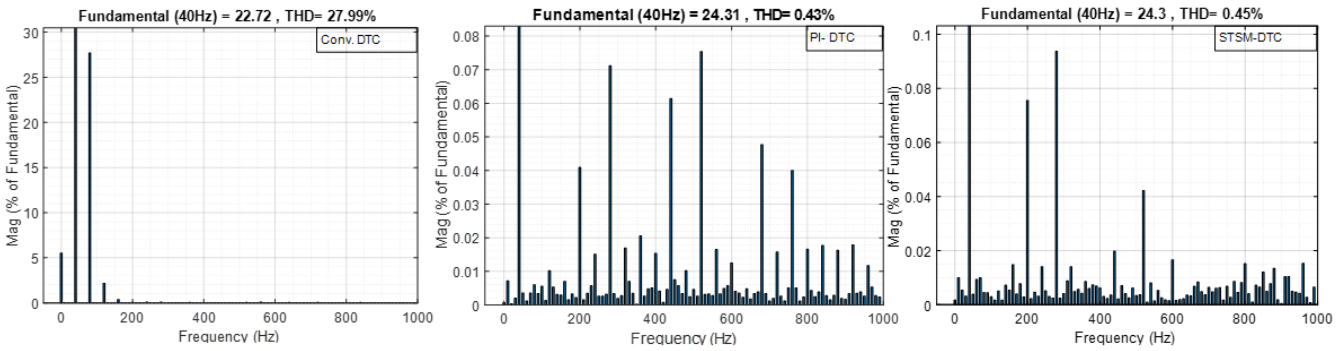


Fig. 21. FFT analysis and spectrum of THD for stator phase current i_{sa} .

According to the above statistics and analysis, it can be concluded that the modified DTC (PI-DTC and STSM-DTC) can obtain better performances in both torque and flux responses. The proposed STSM-DTC performs best in the torque-ripple reduction with $0.0235\ Nm$, relatively fast in dynamic response, but the flux ripple is slightly bigger than that of PI-DTC approach. For DTC drive, torque-ripple reduction is more important; hence, it can be deduced that the STSM-DTC algorithm obtains the best overall torque performances among the three kinds of DTCs for IPMSM.

Moreover, the proposed method can also enhance the speed performances and improve current control with very low THD.

4.2 Simulation verification of NN-PI speed controller

The simulation works of this section are developed to demonstrates the effectiveness and robustness of proposed NN-PI speed controller based STSM-DTC for IPMSM system, as shown in Fig. 22. The motor parameters are kept same to that in section 4.1, as shown in Table 2. For presenting the performances of proposed scheme, the classical PI speed controller based on the same system is also simulated as a comparison.

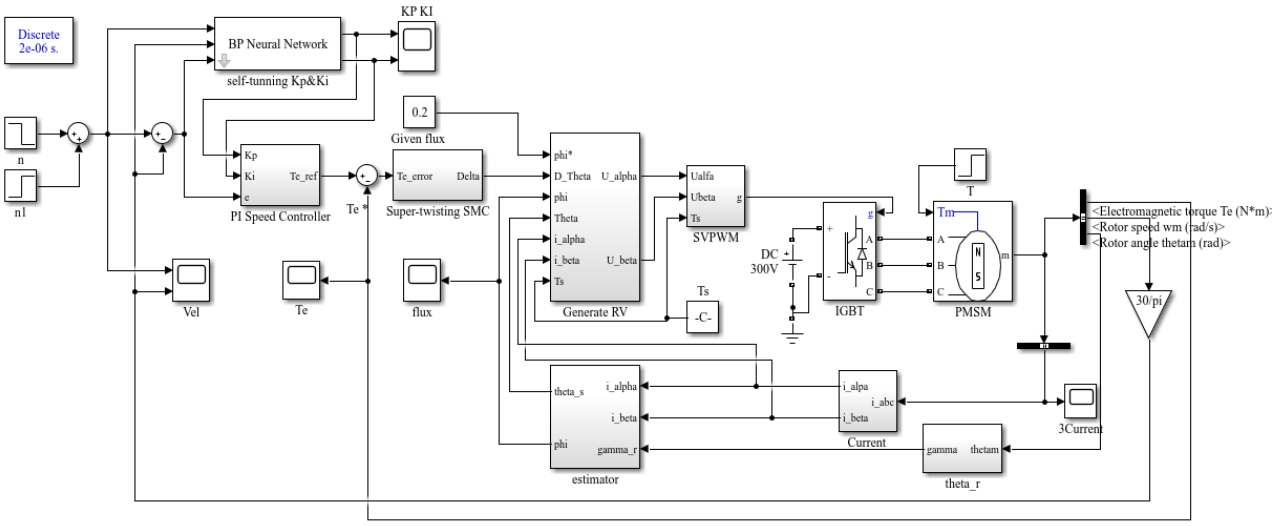


Fig. 22. Simulation block of proposed control scheme for IPMSM.

These two speed control schemes are investigated by subjecting the IPMSM drives to a variable speed operation. Simulation is started at the speed 1200 rpm with a sudden speed reduction to 1000 rpm at 0.1s and a sudden speed increase to 1200 rpm at 0.2s. The load torque is suddenly rising from 0 $N \cdot m$ to 2 $N \cdot m$ at 0.3s. J is set to 0.0008 $kg \cdot m^2$, where the inertia ratio is 1.The learning rate η and momentum coefficient α of NN-PI controller are set as 0.5 and 0.05, respectively. The k_p , k_i of contrasted PI controller are set as 0.1 and 20, respectively.

In this paper, the performances of speed control are evaluated by 4-aspects: start-up speed response, steady-state speed response, speed tracking and robustness against load disturbance. Fig. 23. shows the simulation results of two speed controllers. The 4-ZOOMs respond to 4-aspects of speed performances, that's to say S1 describes start-up speed response, S2 describes steady-state speed response, S3

describes speed tracking and S4 describes robustness against load disturbance. All these 4-aspects of speed control performances will be presented and analyzed by statistics and detailed depicts in the following of this section.

The adaptive controller parameters of k_p and k_i of NN-PI speed controller are tuned by BP algorithm automatically according to the variations of speed command, as shown in Fig. 24. It can be clearly observed that the k_p and k_i will be tuned continuously until the motor tracks the reference speed and the tuning work is very fast. It means the proposed NN-PI speed controller could have optimal parameters at most simulation period.

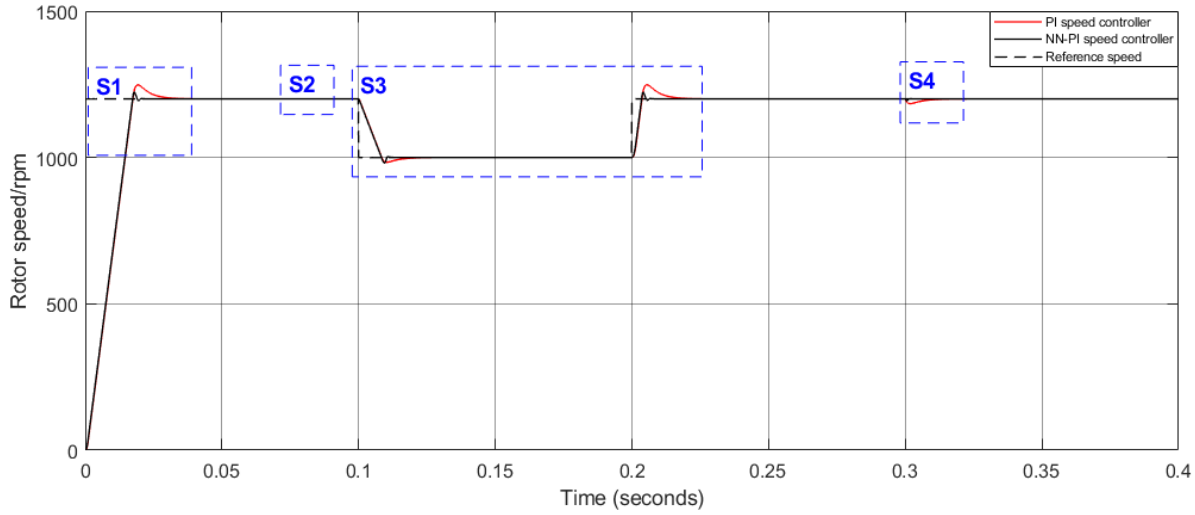


Fig. 23. Speed control performances of PI and NNP-PI controllers.

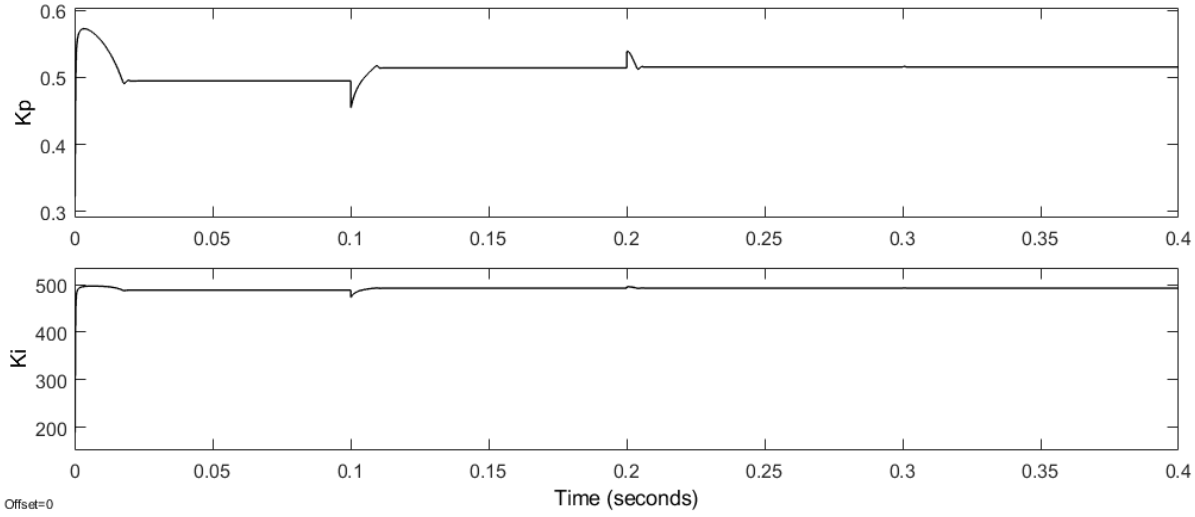


Fig. 24. The auto-tuned parameters k_p and k_i of NN-PI speed controller.

(1) Start-up speed response

The statistics of speed responses in start-up stage S1 of these two methods are shown in Table 7, and the detailed figure is shown in Fig. 25. Obviously, the proposed NN-PI speed controller technique can obtain much better speed performance in start-up stage with only 22.8 *ms* convergence time and 24.2 *rpm* speed overshoot compared to PI speed controller with 45.5 *ms* convergence time and 47.8 *rpm* overshoot. In contrast to classical PI speed controller, the proposed method operates 3 times speed adjusting but has shorter response time and smaller overshoot.

Table 7. The statistics of start-up speed responses in ZOOM-S1

Result-		PI	NN-PI	Unit
Start-up speed response	Convergence time	45.5	22.8	*ms*
[S1]	Overshoot	47.8	24.2	*rpm*

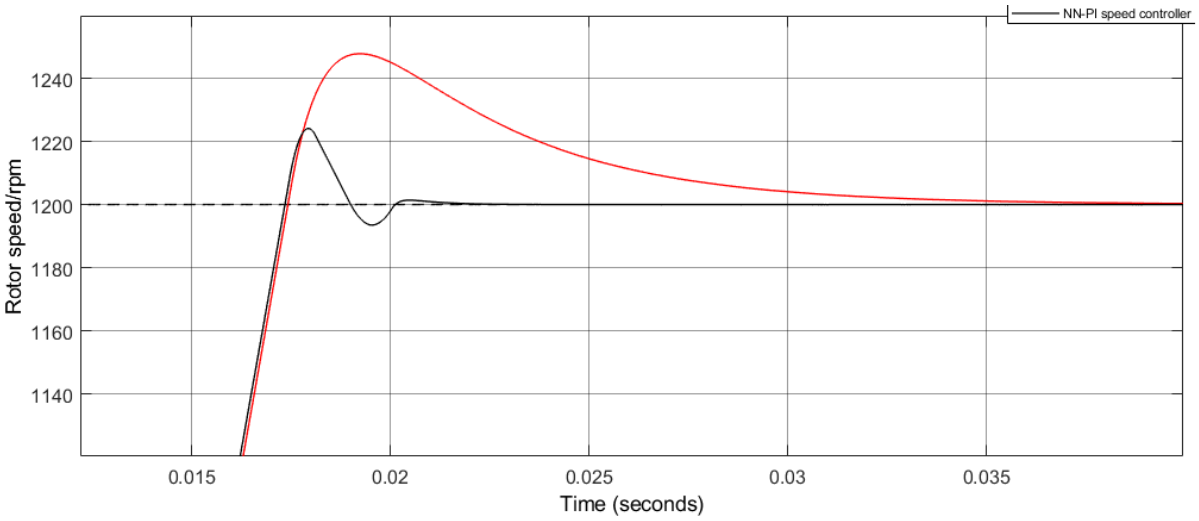


Fig. 25. The detailed figure of start-up speed responses in ZOOM-S1.

(2) Steady-state speed response

The statistics of speed responses in steady-state S2 of these two methods are shown in Table 8, and the detailed figure is shown in Fig. 26. The speed ripples of both methods are very small as the application of proposed STSM-DTC technique. Similarly, the proposed NN-PI speed controller can obtain better speed performance compared to PI speed controller.

Table 8. The statistics of steady-state speed responses in ZOOM-S2

Result-		PI	NN-PI	Unit
Steady speed response	Range	0.059	0.027	*rpm*
[S2]	S-S error	0.024	0.002	*rpm*

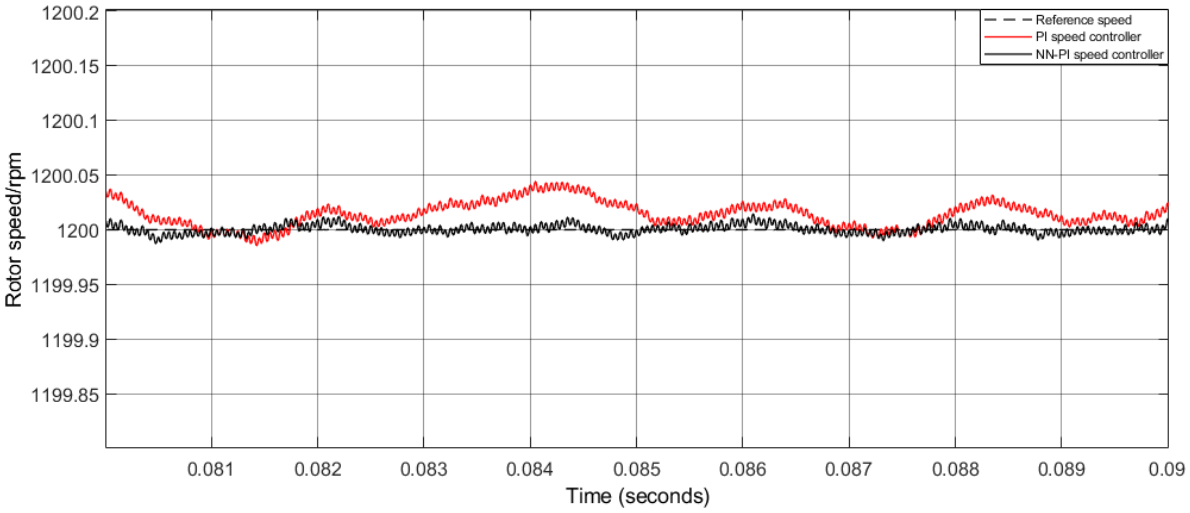


Fig. 26. The detailed figure of steady-state speed responses in ZOOM-S2.

(3) Speed tracking

The statistics of speed tracking in S3 of these two methods are shown in Table 9, and the detailed figure is shown in Fig. 27. It's clear that the proposed NN-PI speed controller technique surpass the classical PI speed controller in speed rising case with smaller speed overshoot. Moreover, the NN-PI speed controller performs much faster in speed dynamic tracking in contrast to the PI speed controller.

Table 9. The statistics of speed tracking in ZOOM-S3

Result-			PI	NN-PI	Unit
Speed tracking	Falling-	Undershoot	16.5	18.9	*rpm*
[S3]		Setting time	35.6	15.1	*ms*
	Rising+	Overshoot	48.6	23.9	*rpm*
		Setting time	23.0	9.1	*ms*

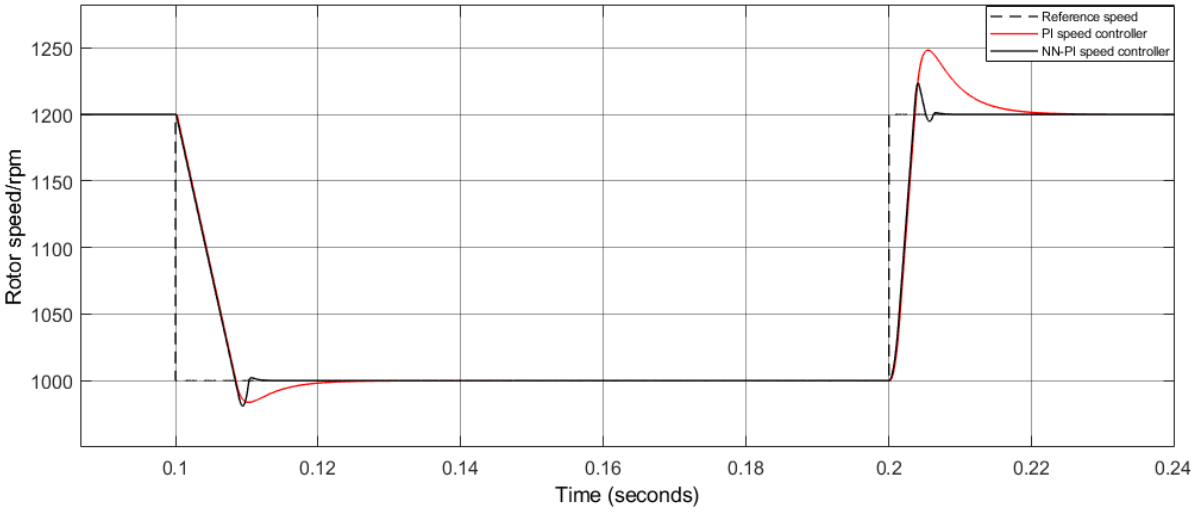


Fig. 27. The detailed figure of speed tracking in ZOOM-S3.

(4) Robustness against load disturbance

The load applied to the shaft of motor can be generally divided into two categories: load torque and load inertia. Both loads may vary unexpectedly in real industrial application, which may diminish the performance of the motor drives. Therefore, the load disturbance will be accordingly divided into load torque disturbance and load inertia disturbance and applied to the tested systems in this part.

Firstly, the J (combined inertia of rotor and load) is kept as constant as $0.0008\,kg\cdot m^2$, which means the applied load inertia J_L is set as same to rotor inertia J_M. Generally, the applied load inertia should less than 5 times to rotor inertia. Under this ideal condition, the influence of load torque disturbance to the systems will be tested as follows.

The emulated statistics of speed responses when load torque disturbance applied S4 of these two methods are shown in Table 10, and the detailed figure is depicted in Fig. 28. As shown that the speed drops are $6.6\,rpm$ and $16.2\,rpm$ with respect to the methods of NN-PI speed controller and PI speed controller; furthermore, the convergence time of NN-PI speed controller is only $4.1\,ms$ compared to that $24.4\,ms$ of PI speed controller. As shown in Fig. 28., the proposed speed controller needs 2 times speed adjusting cycles before converging to command speed but operates very fast.

It's can be concluded that the NN-PI speed controller presents better robustness in the case of load torque variation.

Table 10. The statistics of robustness against load torque variation in ZOOM-S4

Result-		PI	NN-PI	Unit
Robustness [S4]	Setting time	24.4	4.1	*ms*
	Speed drop	16.2	6.6	*rpm*

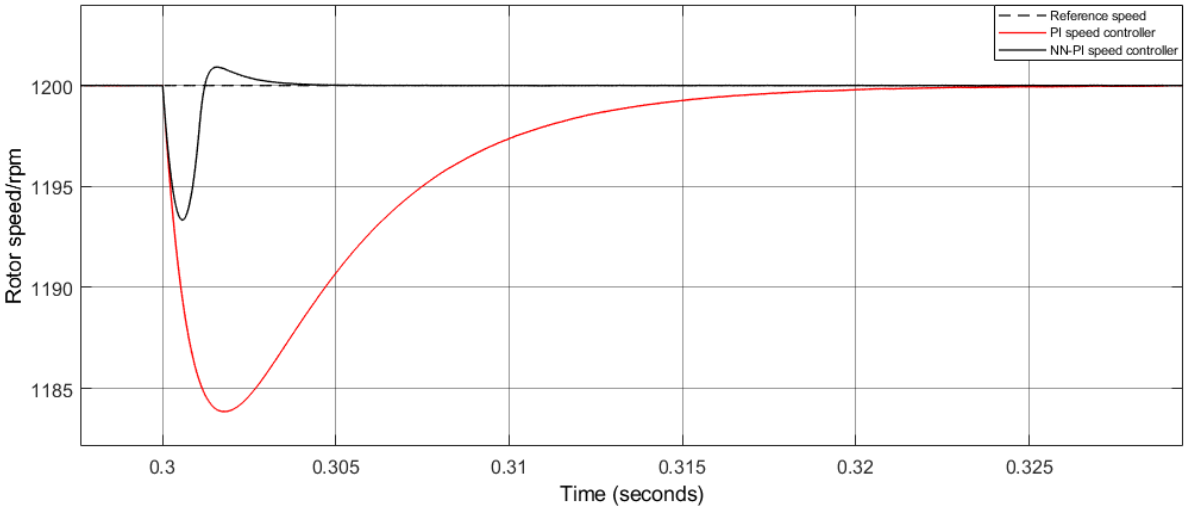


Fig. 28. Speed performance in case of load torque variation in ZOOM-S4.

Next, the applied load inertia will be increased to more than 5 times of rotor inertia, as $0.002\,kg\cdot m^2$, that is $J_L = 5J_M$. The influence of this load inertia variation to proposed NN-PI based STSM-DTC IPMSM drive is simulated and compared to that when $J_L = J_M$. Simulation is started at the speed 1200 rpm with a sudden speed reduction to 1000 rpm at 0.15s and a sudden speed increase to 1200 rpm at 0.3s. The load torque is kept as constant of $1\,N\cdot m$. Except the load inertia is different, the simulation environments and system parameters are keep same.

The simulation result of speed performance comparison in the case of $J_L = J_M$ and $J_L = 5J_M$ is depicted in Fig. 29. As shown, the speed response has a longer convergence time ($69.5\,ms$) but smaller overshoots when speed command varies in the case of $J_L = 5J_M$. It's reasonable since the system inertia J can influence the speed rate of motor as shown in the equation of (1-7). The speed performance is not diminished seriously as presented in [17], which verifies the robustness against the load inertia

disturbance of proposed NN-PI speed controller based STSM-DTC technique.

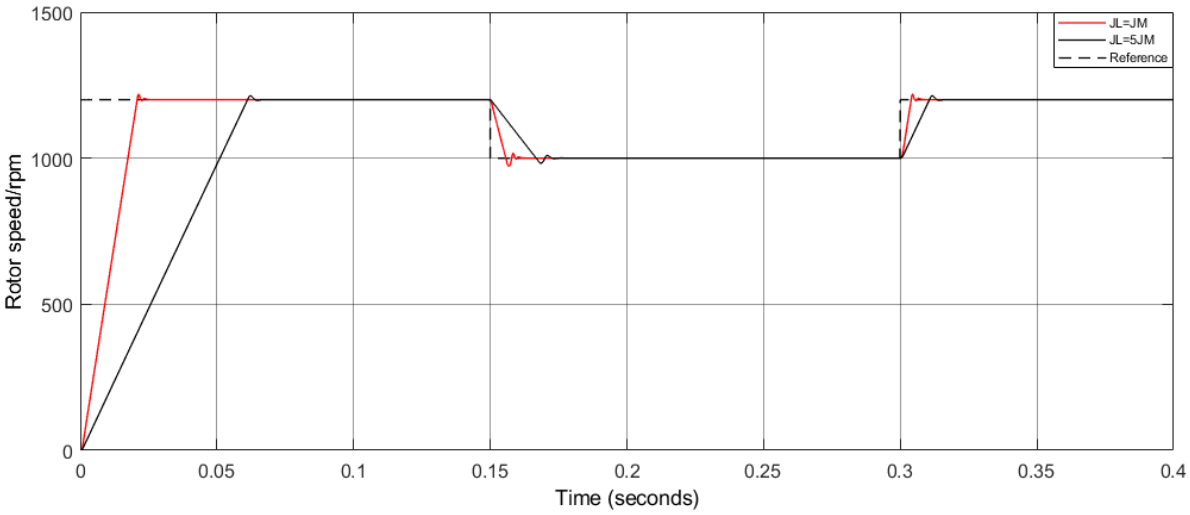


Fig. 29. Speed performance in case of load inertia variation.

Upon the above simulation results and comprehensive analysis, it can be concluded that the proposed NN-PI speed controller can obtain excellent speed performances in both dynamic and steady-state speed responses with smaller overshoot and undershoot, smaller speed ripple, smaller speed drop, and shorter convergence time. In addition, the NN-PI speed controller enhances the robustness against load disturbance in both cases of load torque variation and load inertia variation.

It can be concluded that the proposed NN-PI speed controller is an effective, robust and excellent control method when applying to IPMSM drives.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion of proposed NNPI-STSM-DTC for IPMSM

There are two contributions to the DTC scheme for IPMSM in this thesis. One is the modified DTC algorithm by using super-twisting sliding mode control algorithm, named as STSM-DTC. The other is the parameters self-tuning speed controller through backpropagation based neural network, named as NN-PI speed controller. Both proposed techniques applied for IPMSM drives are demonstrated by simulation work and comparison with other methods.

Comparative results in section (4-1) certainly indicate that the proposed STSM-DTC method can enhance the performance considerably with very faster torque response, smaller torque ripple as about 0.1Nm and relatively smaller flux ripple as about 0.0004Wb, as shown in Table 5,6. Moreover, it has smaller torque overshoot during load torque disturbance, as shown in Table 5, compared to other DTC schemes. We also find that It can enhance the speed performance with smaller speed ripple. In addition, through FFT analysis, it can be seen the proposed STSM-DTC strategy can obtain good performance of stator current with lower THD (total harmonic distortion) level as 0.45%.

Furthermore, due to the application of neural network, the proposed NN-PI speed controller can improve the speed control a lot as it has the adaptive tuning parameters of controller against to speed command variation and load disturbance. The simulation work designed in Chapter 4 and results can fully demonstrate the effectiveness of NN-PI speed controller. It has the merits of very fast dynamic response with very small convergence time, very small overshoot and speed drop, and excellent robustness against load disturbance in both cases of load torque variation and load inertia variation.

5.2 Future work

The proposed NNPI speed controller based STSM-DTC control strategy for IPMSM drives is only tested and proved by simulation, which means all environment and condition settings are ideal. In the future, the proposed strategy should be verified by experiment using HIL (hardware in loop) method through dSPACE software.

Moreover, in practical industrial using, the motor parameters will vary because of complex dynamic environment, which also decrease the application of proposed DTC method. The stator resistance varies when environment temperature changes, which affects the motor control. In addition, the induction parameters variation also harms to control of PMSM drives. The future work will focus on motor parameters observer design based sliding mode control technique and improve the "intelligence" of NN-PI speed controller against the parameters variations. As shown in section (4-2), when the load inertia becomes bigger, the speed convergence time will be longer. The state observer of load disturbance is necessary, which can provide feedforward information to NN-PI speed controller. Therefore, the speed performance can be improved more and obtain better robustness against load disturbances.

Additionally, the proposed NN-PI speed controller may be improved by using RPROP (resilient backpropagation) algorithm, which is verified effectively in [16], [29].

[REFERENCE]

[1] E. Schaltz, "Electrical vehicle design and modeling," in: *S. Soylu (Ed.), Electric Vehicles Institute Modelling and Simulations,* InTech, Croatia, pp. 1–24, 2011.

[2] Song, Q.; Li, Y.; Jia, C, "A Novel Direct Torque Control Method Based on Asymmetric Boundary Layer Sliding Mode Control for PMSM," *Energies*, vol. 11, no. 3, pp. 657-672, Mar. 2018.

[3] Ke Song, Wei guo Liu. "Permanent Magnet Synchronous Motor Field Oriented Control and HIL Simulation," *IEEE Vehicle Power and Propulsion Conference,* Harbin, pp. 4-6, 2008.

[4] C. Xia, S. Wang, X. Gu, Y. Yan, and T. Shi, "Direct torque control for VSI-PMSM using vector evaluation factor table," *IEEE Trans. Ind. Electron.,* vol. 63, no. 7, pp. 4571-4583, July 2016.

[5] A. Haddoun, M.E.H Benbouzid, D. Diallo, R. Abdessemed, J. Ghouili, K.Srairi, "A loss-minimization DTC scheme for EV induction motors," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 81–88, Jan. 2007.

[6] I. Takahashi, T. Noguchi, "A new quick-response and high-efficiency control strategy of an induction motor," in *conf. Rec. IEEE IAS Annu. Meeting*, 1985, pp. 1665-1675.

[7] Q. Liu and K. Hameyer, "Torque ripple minimization for direct torque control of pmsm with modified fcsmpc," *IEEE Trans. Ind. Appl.*, vol. 52, no. 6, pp. 4855–4864, Nov/Dec 2016.

[8] C. A. Martins, X. Roboam, T. A. Meynard, and A. S. Carvalho, "Switching frequency imposition and ripple reduction in DTC drives by using a multilevel converter," *IEEE Trans. Power Electron.*, vol. 17, no. 2, pp. 286–297, Jan. 2002

[9] A. Ammar, A. Bourek, and A. Benakcha, ''Nonlinear SVM-DTC for induction motor drive using input-output feedback linearization and high order sliding mode control,'' *ISA Trans*., vol. 67, pp. 428–442, Mar. 2017.

[10] Y. Choi, H. Choi, and J. Jung, "Feedback linearization direct torque control with reduced torque and flux ripples for IPMSM drives," *IEEE Trans. Power Electron*., vol.31, no. 5, pp. 3728-3737, May. 2016.

[11] K. Chikh, M. Khafallah and A. Saad, "Improved DTC Algorithms for Reducing Torque and Flux Ripples of PMSM Based on Fuzzy Logic and PWM Techniques", *MATLAB- A Fundamental Tool for Scientific Computing and Engineering Applications*, Vol. 1, pp. 183-189, 2012.

[12] L. Tang, L. Zhong, M. F. Rahman, and Y. Hu, “A novel direct torque controlled interior permanent magnet synchronous machine drive with low ripple in flux and torque and fixed switching frequency,” *IEEE Trans. Power Electron.*, vol. 19, no. 2, pp. 346–354, Mar. 2004.

[13] Y. Cho, K.-B. Lee, J.-H. Song, and Y.-I. Lee, “Torque-ripple minimization and fast dynamic scheme for torque predictive control of permanent magnet synchronous motors,” *IEEE Trans. Power Electron*., vol. 30, no. 4, pp. 2182–2190, Apr. 2015.

[14] Y. Su, C. Zheng, and B. Duan, “Automatic disturbances rejection controller for precise motion control of permanent-magnet synchronous motors,” *IEEE Trans. Ind. Electron.*, vol. 52, no. 3, pp. 814–823, Jun. 2005.

[15] W. Yu, G. Liang, W. Hao, and W. Xiao, “Control method for optimal dynamic performance of DTC based PMSM drives”, *IEEE Trans. Energy Conver.*, Jan. 2018.

[16] T. Pajchrowski, K. Zawirski, and K. Nowopolski, “Neural speed controller trained online by means of modified RPROP algorithm,” *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 560–568, Apr. 2015.

[17] S. Li and Z. Liu, “Adaptive speed control for permanent magnet synchronous motor system with variations of load inertia,” *IEEE Trans. Ind. Electron*., vol. 56, no. 8, pp. 3050–3059, Aug. 2009.

[18] J. Jung, V. Q. Leu, T. D. Do, E. K. Kim, and H. H. Choi, “Adaptive PID speed control design for permanent magnet synchronous motor drives,” *IEEE Trans. Power Electron*., vol. 30, no. 2, pp. 900–908, Feb. 2015.

[19] H. Abu-RuB, A. Iqbal, and J.Guzinski, “High performance control of AC drives with MATLAB/Simulink models”, *John Wiley & Sons Ltd*, pp.173-175, 2012.

[20] C. Gershenson, "Artificial neural networks for beginners," *arXiv preprint*, Sept. 2003.

[21] A Krenker, J Bešter and A Kos, “Introduction to the Artificial Neural Networks”, Edited Kenji Suzuki, Published by *InTech*, Janeza Trdine, Croatia, pp 6-7, 2011.

[22] Michael A. Nielsen, “Neural Networks and Deep Learning”, *Determination Press*, Chap2, 2015.

[23] J. Kang, W.J. Meng, A. Abraham, H.B. Liu, “An adaptive PID neural network for complex nonlinear system,” *Neurocomputing*, vol. 135, pp.79–85, 2014.

[24] X. Fu and S. Li, ‘‘A novel neural network vector control technique for induction motor drive,’’ *IEEE Trans. Energy Convers.*, vol. 30, no. 4, pp. 1428–1437, Dec. 2015.

[25] C. Lascu, I. Boldea, and F. Blaabjerg, “Super-twisting sliding mode control of torque and flux in permanent magnet synchronous machine drives,” in Proc. *IEEE IECON*, pp. 3171–3176, 2013.

[26] A. Levant, “Higher-order sliding modes, differentiation and output feedback control,” *Int. J. Control*, vol. 76, no. 9/10, pp. 924–941, 2003.

[27] Zhenyu Jia and Byeongwoo Kim, “Online trained neural network-PI speed controller for DTC based IPMSM drives,” *International Journal of Electrical and Electronic Engineering & Telecommunications,* Vol. 7, no. 3, to be published in July, 2018.

[28] Zhenyu Jia and Byeongwoo Kim, “Direct torque control with adaptive PI speed controller based on neural network for PMSM drives,” in Proc. MATEC Web Conf., vol. 160, Apr. 2018.

[29] L. Grzesiak, M. Meganek, J. Sobolewski, and B. Ufnalski, “On-line trained neural speed controller with variable weight update period for direct-torque-controller AC drive,” in Proc. *EPE-PEMC Conf.*, Portoroz, pp. 1127–1132, 2006.

[APPENDIX]

The programming work of backpropagation neural network based PI speed controller is realized by s-function on Matlab. The main coding is as follows:

```
function [sys,x0,str,ts]=BPNN(t,x,u,flag,T,nh,xite,alfa)
switch flag
   case 0
       [sys,x0,str,ts] = mdlInitializeSizes(T,nh);
   case 3
       sys = mdlOutputs(t,x,u,T,nh,xite,alfa);
    case {1, 2, 4, 9}
        sys = [];
    otherwise
       error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes(T,nh)
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 3+5*nh;
sizes.NumInputs = 6+10*nh;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [T 0];
function sys = mdlOutputs(t,x,u,T,nh,xite,alfa)
     if u(4)<=0
        wi_2=0.5*rands(nh,3); %% give initial weights
        wo_2=0.5*rands(2,nh);
        wi_1=0.5*rands(nh,3);
        wo_1=0.5*rands(2,nh);
     else
wi_2=reshape(u(8:7+3*nh),nh,3); %% training process: updating weights
wo_2=reshape(u(8+3*nh:7+5*nh),2,nh);
wi_1=reshape(u(8+5*nh: 7+8*nh),nh,3);
wo_1=reshape(u(8+8*nh:7+10*nh),2,nh);
     end
```

```
u(1)=u(1)/u(5); %% Sample normalization, inputs associated to reference n
u(2)=u(2)/u(5);
u(3)=u(3)/u(5);
u(4)=u(4)/u(5);
u(5)=u(5)/u(5);
xi=[u([5,3,1])']; % inputs of network xi=[r,y,e]
xx=[u(1)-u(2); u(1)];
I=xi*wi_1'; %% the input of hidden layer (z(2))
Oh=non_transfun(I,1); %% the output of hidden layer (a(2))
K=non_transfun(wo_1*Oh',2);%the output of output layer(k=(kp;ki))
K(1)=K(1); %% kp
K(2)=K(2)*500;%% ki with coefficient
uu=u(6)+[K(1), K(2)]*xx; %%the output of incremental PI u(k) of inner loop
dyu=sign((u(3)-u(4))/(uu-u(6)+0.0000001));
dK=2*non_transfun(wo_1*Oh',3); %% g'(netki) NOTICE!
delta3=u(1)*dyu*xx.*dK;
wo=wo_1+xite*delta3*Oh+alfa*(wo_1-wo_2);% weights update of output layer
dO=4*non_transfun(I,3);
wi=wi_1+xite*(dO.*(delta3'*wo))'*xi+alfa*(wi_1-wi_2);%weights update of
hidden layer
sys=[uu;K(1);K(2);wi(:);wo(:)];%outputs of BPNN system
%%%%%% types of activation function %%%%%%%%%%%%%%%%%%
function W1=non_transfun(W,key)
switch key
   case 1, W1=(exp(W)-exp(-W))./(exp(W)+exp(-W));
   case 2, W1=exp(W)./(exp(W)+exp(-W));
   case 3, W1=1./(exp(W)+exp(-W)).^2;
end
```