Neural Network-Based Self-triggered Attitude Control of a Rigid Spacecraft

Shuai Sun, Mengfei Yang and Lei Wang

1 Introduction

Modern spacecrafts are typical Cyber-Physical Systems (CPSs) which are tight integrations between onboard cyber systems (e.g. processing, communication) and physical elements (e.g. platform structure, sensing, actuation, and environment) [1, 2]. Tasks implement on the onboard cyber systems of spacecrafts include attitudes and orbit control, onboard planning and scheduling, onboard data acquiring and analyzing et al. For the limit of onboard digital resource, saving communication bandwidth and power from the basic function like control tasks mean providing more resource to payloads onboard, which may implement a sufficient use of spacecrafts.

Classic spacecraft attitude control problem is always developed in continuous framework [3–5], and their implementation on the digital platform is traditionally in a periodic fashion due to the ease of design and analysis. However, periodic sampling is sometimes less preferable from a resource utilization point of view. When the system states almost keep constant and no disturbances are acting, the systems are operating desirably is a waste of digital resources. To overcome these limitations, some resource-aware control laws like anytime control [6] event-triggered control (ETC) [7–9] self-triggered control (STC) [10, 11] and so on are developed. In anytime control, consisting in a hierarchy of controllers for the same plant, will switch between different order control algorithms to adopt the cyber resource of the onboard system. In ETC, the classic event triggered mecha-

S. Sun (✉) · L. Wang
Beijing Institute of Control Engineering, Beijing 100081, China
e-mail: sunshuai_hitsa@163.com

L. Wang
e-mail: Wangl01@mails.tsinghua.edu.cn

M. Yang
China Academy of Space Technology, Beijing 100190, China
e-mail: yangmf@bice.org.cn

nisms are detecting the system output or states varying. The main idea of STC is to select the next controller update instant based on the knowledge of the dynamics and the latest measurements of the plant state. While the ETC depending on the continuous supervision of the plant, the STC will reduce more cyber resource than ETC theoretically.

In recent years, many work addressed on STC and its applications in CPSs. In [9], the theoretical foundation rendering control system Input-to-State Stable (ISS) is introduced. Based on this framework, many STC mechanisms are proposed for linear and nonlinear systems. In [10–15], the basic self-triggered strategies for linear system are presented and a trade-off between performance and cyber resource utilization is making. However, the most expensive part of the realization of STC is the derivation of the trigger function. The above work is extremely complex so that the calculating of the trigger functions requires a lot of computing resources. From this point of view, the use of NN as a model of trigger behavior is presented in this paper.

The Neural Network is usually performed to estimate the system that transforms inputs into outputs, where a set of examples of input-output pairs are given in [16]. Furthermore, the NN was demonstrated as a universal smooth function approximator, extensive studies have been conducted for diverse applications, especially pattern recognition, identification, estimation, and time series prediction [17]. In [18], the NN is designed to predict the time delay induced in the networked control systems, which shows that the NN can alleviate the influence of time delay and improve the performance of the networked control system.

The main contribution of this paper is the design of a neural network based self-triggered control (NN-STC) strategy for the spacecraft attitude stabilization problem. The attitude motion dynamics equations are derived and a STC algorithm is proposed to ensure attitude stability. Furthermore, simulation results demonstrate the proposed approach guarantees a high control performance and a large reduction of sampling times.

2 Space Attitude Dynamics

The dynamics equations of motion of a rigid spacecraft are given by [4]

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}^{\times}\mathbf{J}\boldsymbol{\omega} + \boldsymbol{u} + \boldsymbol{T} \tag{1}$$

where $\mathbf{J} = \mathbf{J}^{T} \in \mathbb{R}^{3\times 3}$ is the inertia matrix, $\boldsymbol{\omega} = \boldsymbol{\omega}(t) \in \mathbb{R}^{3}$ denotes the body angular velocity of the spacecraft body-fixed frame B with respect to the inertial frame I. $\boldsymbol{u} \in \mathbb{R}^{3}$ is the actuator torque calculated by the onboard computer and $\boldsymbol{T} \in \mathbb{R}^{3}$ is the external disturbance torques.

To describe the orientation of B respect to the inertial frame I in terms of three Euler angles: roll angle φ, pitch angle θ, and yaw angle ψ. In the rigid body

three-axis stability control problem, the Euler angles are all small. Therefore, the linearized kinematic differential equations can be found as

$$\omega_1 = \dot{\varphi} - \omega_o\psi, \quad \omega_2 = \dot{\theta} - \omega_o, \quad \omega_3 = \dot{\psi} + \omega_o\varphi \tag{2}$$

We consider the attitude motion about its pitch axis, assume the effect of the attitude motion on the orbital motion with a constant angular velocity ω_o.

In such case, the dynamics equations (1) with small attitude errors and small products of inertial become [19]

$$J_x\ddot{\varphi} + (J_y - J_z)\omega_o^2\varphi + (J_y - J_z - J_x)\omega_o\dot{\psi} = u_x + T_x, \quad J_y\ddot{\theta} = u_y + T_y$$
$$J_z\ddot{\psi} + (J_y - J_z)\omega_o^2\psi - (J_y - J_z - J_x)\omega_o\dot{\varphi} = u_z - \omega_o h_x + T_z \tag{3}$$

Also, the external disturbance torques include the gravity gradient torque, and it can be expressed in the body frame B as

$$T_{gx} \approx -3\omega_o^2(J_y - J_z)\varphi, \quad T_{gy} \approx -3\omega_o^2(J_x - J_z)\theta, \; T_{gz} \approx 0 \tag{4}$$

Generally, from the equations above, we can see that the pitch equation is uncoupled from the roll/yaw equations. It is a Single Input & Single Output system (SISOs), which is easy to design a classical PD controller. However, if we take three dynamics equations into consideration with the self-triggered control design, the size of the state space is large, which will provide more pressure on the real time calculation on the embedded system. Therefore, we only consider roll & yaw coupled system in this paper.

The dynamics and kinematic equations can be realized in state space form as

$$\dot{x} = Ax + Bu, \quad x = \begin{pmatrix} \varphi & \psi & \dot{\varphi} & \dot{\psi} \end{pmatrix}^T \tag{5}$$

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{1}{J_{11}}a & 0 & 0 & \frac{1}{J_{11}}b \\ 0 & -\frac{1}{J_{33}}c & \frac{1}{J_{33}}b & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{J_{11}} & 0 \\ 0 & \frac{1}{J_{33}} \end{pmatrix}$$

3　Neural Network Based Self-triggered Control Strategy Design

3.1　Problem Formulation

The attitude control system on the spacecraft is shown in Fig. 1. The control signal u and states signal $(x, \dot{x})$ is transferred over the network. In a periodic implementation, this system is usually in the following form
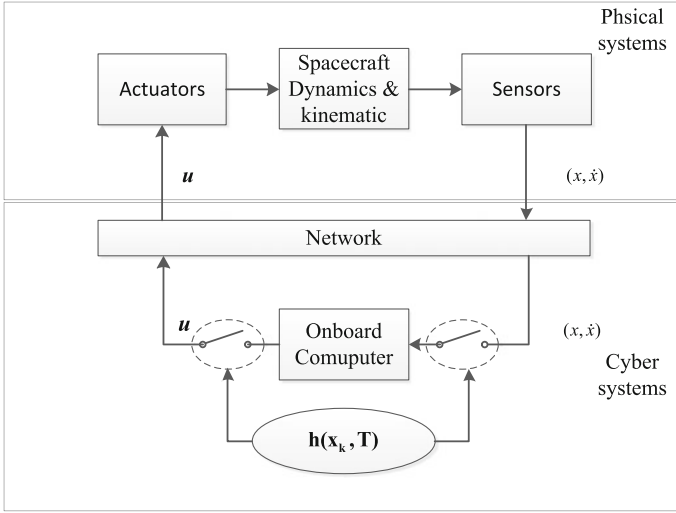
Fig. 1 CPSs on the spacecrafts

$$\dot{x} = \mathbf{A}x(t) + \mathbf{B}u(t)$$
$$y(t) = \mathbf{C}x(t) \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times r}$, $\mathbf{C} \in \mathbb{R}^{m \times n}$ are the characteristic matrices and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^r$ and $y(t) \in \mathbb{R}^m$ the state, input and output vectors respectively. If the pair $(\mathbf{A}, \mathbf{B})$ is stabilizable, then a linear feedback controller $K : \mathbb{R}^n \to \mathbb{R}^m$ rendering the closed-loop asymptotically stable can be found:

$$u(t) = K(x(t)) \tag{7}$$

where $\mathbf{A} + \mathbf{B}\mathbf{K}$ is Hurwitz.

The controller will implement on the embedded system, and a sampled-data system implementation is in a classical periodic time-triggered way:

$$\dot{x} = \mathbf{A}x(t) + \mathbf{B}u(t_k) \quad t \in [t_k, t_{k+1})$$
$$y(t) = \mathbf{C}x(t) \tag{8}$$
$$u(t) = K(x(t_k)), \quad t \in [t_k, t_{k+1}), \ k \in \mathbb{Z}_+$$

where t_k represents the sampling time satisfying $t_{k+1} - t_k = T$. For some specified period $T > 0$, it means $h(x_k, T)$ in Fig. 1 equals a constant T. In discrete control aspect, T is generally chosen as small as technology and network load permit to achieve a desired performance. This strategy is easy to implemented, however, it may produce unnecessary overload on the network and power consumption.

We design the Performance Index (PI) ΔJ

$$\Delta J = 100 \left[\int_0^\infty \left(\boldsymbol{x}^T \mathbf{Q}_{lqr} \boldsymbol{x} + \boldsymbol{u}^T \mathbf{R}_{lqr} \boldsymbol{u} \right) dt - J^* \right] \tag{9}$$

when any control $\boldsymbol{u}(t)$ is implemented digitally, the system performance will depend on the sampling period. As the sampling period increasing, the PI will increase and the system tends to unstable.

To extend the sampling periods without sacrificing performance, we apply self-triggered control to the system. Moreover, the longer mean sampling period means the lower network loaded.

The STC scheme can be defined in following form

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t_k) \quad t \in [t_k, t_{k+1}) \\
\boldsymbol{y}(t) &= \mathbf{C}\boldsymbol{x}(t) \\
\boldsymbol{u}(t_k) &= \boldsymbol{K}(\boldsymbol{x}(t_k)), \quad t \in [t_k, t_{k+1}), \ k \in \mathbb{Z}_+ \\
t_{k+1} &= t_k + h(\boldsymbol{x}_k)
\end{aligned} \tag{10}$$

A general STC scheme usually means two functions: A feedback control $\boldsymbol{u}(t_k)$ is used as in the classical frame and the triggered function $h(\boldsymbol{x}(t_k), T)$: $\mathbb{R}^m \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$, determining the sampling time t_{k+1} as a function of the state $\boldsymbol{x}(t_k)$ at the time t_k. The design of these two functions is the main problem in the STC study. Moreover, a positive minimal inter-sample time is also required to fulfill, to avoid the Zeno phenomenon.

3.2 Feedback Controller Design

An LQR design technique is usually applied to the spacecraft attitude control problem. The cost function and weighting matrices used in the LQR design are

$$J^* = \min(J(\boldsymbol{u})) = \int_0^\infty \left(\boldsymbol{x}^T \mathbf{Q}_{lqr} \boldsymbol{x} + \boldsymbol{u}^T \mathbf{R}_{lqr} \boldsymbol{u} \right) dt \tag{11}$$

$$\mathbf{Q}_{lqr} = \begin{bmatrix} \mathbf{I}_{2\times2} & \\ & 5\mathbf{I}_{2\times2} \end{bmatrix} \quad \mathbf{R}_{lqr} = 10\mathbf{I}_{2\times2}$$

The consideration of selecting weighting matrices in this paper is increasing system dumpling and preventing the actuators saturation.

we can solve the Riccati equation to get the unique positive definite solution $\mathbf{P}$

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}^{-1}\mathbf{R}_{lqr}\mathbf{B}^T\mathbf{P} + \mathbf{Q}_{lqr} = \mathbf{0} \tag{12}$$

The optimal feedback policy K is given by

$$K = -R_{lqr}^{-1}\mathbf{B}^T\mathbf{P} \tag{13}$$

The true cost of continuous LQR is

$$J^* = \mathbf{x}_0^T\mathbf{P}\mathbf{x}_0 \tag{14}$$

3.3 NN Based Triggered Condition Design

Under certain conditions, it has been proved that the neural networks have function approximation abilities and have been frequently used as function approximators. In this section, the NN is used to approximate the function $h(\mathbf{x}_k, T)$ in Fig. 1, $h(\mathbf{x}_k, T)$ determines the next sampling time based on the states sampled before, so it is a mapping between state space and time space. And a neural network model can be derived from measured input/output data of the original triggered conditions.

As one of the classical neural networks, BP feedforward neural network has a simple topology and strong generalization ability. We use a four-layered feedforward neural network and adopt the BP learning algorithm to determine the sampling time.

Figure 2 shows the specific configuration of the BP feedforward network for approximating the triggered function. The BP network consists of four layers: input layer with four neurons presenting the four input states of dynamics systems (5). The neuron numbers of the two hidden layers are 12 and 4, which greatly improve the accuracy of the network. The output layer which has only one neuron for the triggered time is one dimension. The number of hidden layers and neurons are varied, but in this problem the above choice is enough.

Fig. 2 BP network configuration for triggered time prediction
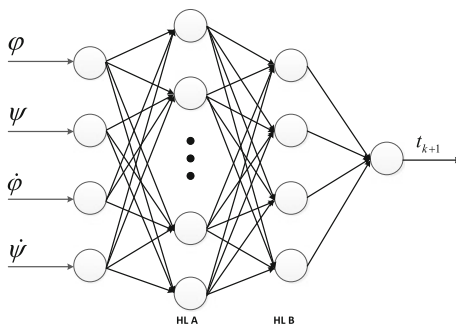
In order to make the NN perform the desired mapping performance, the connection weights will be trained by so-called training algorithms. The detail of this algorithm is given in [20] which is no longer mentioned here. Moreover, the training data generation is another important issue.

We acquire a training set which is generated by the system under closed-loop and the specified triggered condition. The input sets with four states $(\varphi, \psi, \dot{\varphi}, \dot{\psi})$ are generated randomly within $(-5, 5)$ deg and $(-5, 5)$ deg/s which ensure the linearized model is effective. And a triggered time is calculated from a input sets by the original triggered function. Finally, a training sample with five elements $(\varphi, \psi, \dot{\varphi}, \dot{\psi}, t_k)$ is acquired.

4 Simulation

In this section, an example is presented to validate the proposed STC strategy for spacecraft attitude control problem. The inertia matrix of spacecraft **J** is assumed to be as follows

$$\mathbf{J} = \begin{bmatrix} 10.5 & & \\ & 8 & \\ & & 6.75 \end{bmatrix} \text{kg m}^2$$

The control authority is limited by the maximum torque provided by the actuators is ± 0.15 N m, orbit angle velocity is $\omega_o = 0.0612$ deg/s.

The time of simulation is 80 s, the initial attitude and angular velocity is

$$\begin{bmatrix} \varphi_0 & \psi_0 \end{bmatrix}^T = \begin{bmatrix} 5 & -3 \end{bmatrix}^T \text{deg}, \quad \begin{bmatrix} \dot{\varphi}_0 & \dot{\psi}_0 \end{bmatrix}^T = \begin{bmatrix} 2 & 1 \end{bmatrix}^T \text{deg/s}$$

The disturbance is given by:

$$\begin{pmatrix} d_{ex} \\ d_{ez} \end{pmatrix} = \begin{pmatrix} 0.0015 + 0.0012 \sin(0.263t) \\ 0.0015 + 0.0012 \sin(0.245t) \end{pmatrix}$$

The triggered function is given by

$$t_{k+1} = \sup\{t < t_{\max} | h(\boldsymbol{x}(t_k), t) < 0\}$$
$$h(\boldsymbol{x}(t_k), t) := \boldsymbol{x}(t_k + t)^T \mathbf{P} \boldsymbol{x}(t_k + t) - \boldsymbol{x}(t_k)^T \mathbf{P} \boldsymbol{x}(t_k)^T \exp(-\lambda t)$$

P is calculated by (14), $\lambda = 1.0524$, the detail is given in [9–11].

From Table 1, we have compared the performance between the NN with one and two hidden layers. The results show the two hidden layers of NN has better

Table 1 Performance of the NN

Two hidden layers (neurons: 12.4)		One hidden layer (neurons: 18)	
Performance (MSE)	Iterations	Performance (MSE)	Iterations
0.01 (t1)	6	0.01	4
0.001 (t2)	9	0.001	45
0.0005 (t3)	58	0.0005	Failed
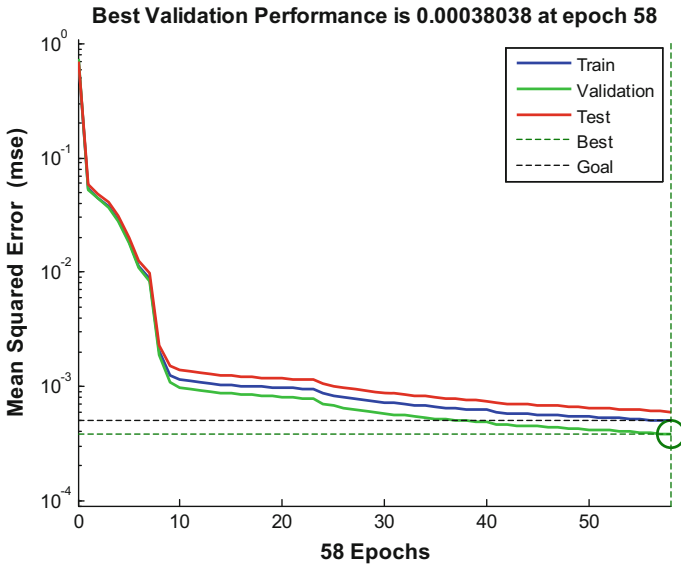
Fig. 3 Training curve of the NN

performance. Figure 3 is the training curve of the NN, it shows that the training progress becomes very slow after 10 iterations.

From Fig. 4, we can see the Performance Index (PI) ΔJ is increasing with the sampling period increases in the scheme of periodic time triggered LQR. Moreover, with the performance of NN increasing, the mean sampling times increases in the scheme of the NN-STC. But the PI is much lower than periodic time triggered control with the same mean sampling period. It means the control performance of NN-STC is better than the periodic time triggered control using the same use of network resource.

The attitude angle and angular velocity errors, by using STC and classical nonlinear controller are shown in a, b of Fig. 5. Roughly speaking, the errors converge to a small level. From c of Fig. 5, it can be seen that the sampling interval continuous increases and it keeps constant after 30 s. It means when the states
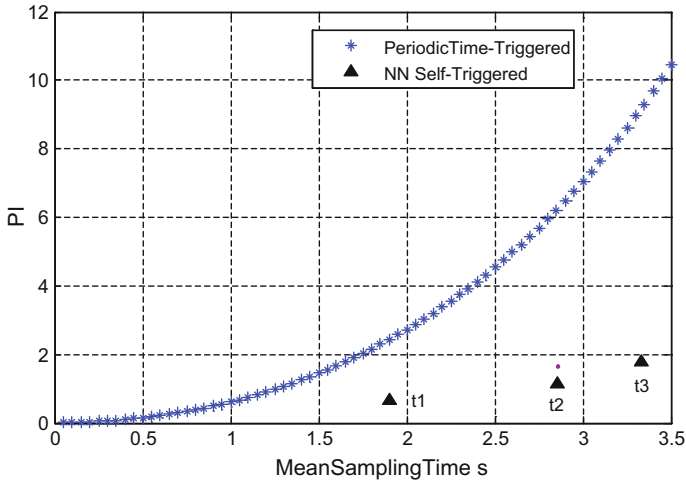
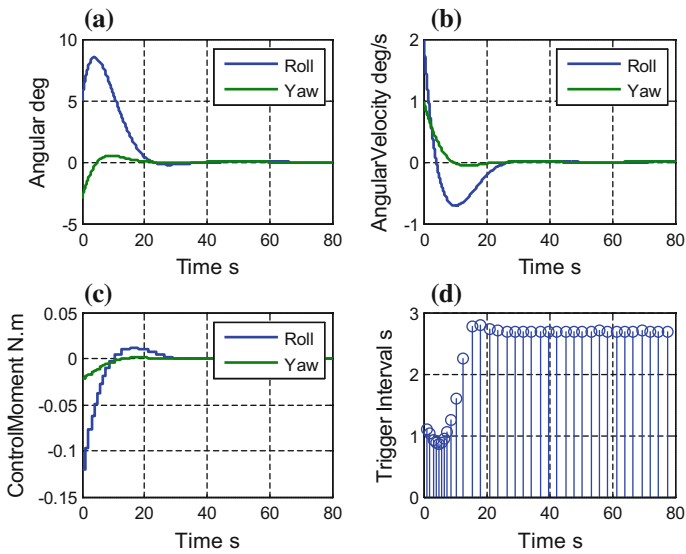**Fig. 4** Performance comparison between PTTC and NN-STC

Fig. 5 States responses and triggered intervals using NN-STC

convince to the steady equilibrium, the controller will entry a salient periods, which is advantageous to save the cyber resources onboard. During 0–10 s, sampling intervals changes with the states overshoot. It means the NN-STC scheme works when the system quits steady process immediately.

5   Conclusion

The main contribution of this paper is the development and implementation of a neural network based self-triggered control for the attitude stabilization of a rigid spacecraft. The control law ensures the system stability of the closed-loop system to the desired attitude. The approach is validated in the simulation and it shows the proposed NN-STC strategy reduces sampling times without degrading the control system performance. The overall performance of the spacecraft is improved by this CPS aspect design.

References

1. Lee EA (2008) Cyber physical systems: design challenges. In: IEEE International symposium on object oriented real-time distributed computing, pp 363–369
2. Yang M, Wang L, Gu B, Zhao L (2012) The application of CPS to spacecraft control system. Aerosp Control Appl 38(5):8–13
3. Tayebi A (2008) Unit quaternion-based output feedback for the attitude tracking problem. IEEE Trans Autom Control 53(6):1516–1520
4. Hu Q (2009) Robust adaptive backstepping attitude and vibration control with L2-gain performance for flexible spacecraft under angular velocity constraint. J Sound Vib 327(3):285–298
5. Wu B (2015) Spacecraft attitude control with input quantization. J Guid Control Dyn 39:1–5
6. Bhattacharya R, Balas GJ (2004) Anytime control algorithm: model reduction approach. J Guid Control Dyn 27(5):767–776
7. Årzén KE (1999) A simple event-based pid controller. In: Proceedings of IFAC world congress
8. Åström KJ, Bernhardsson B (2002) Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: Proceedings of the IEEE conference on decision and control, 2002, vol 2, 2011–2016
9. Tabuada P (2007) Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans Autom Control 52(9):1680–1685
10. Jr MM, Anta A, Tabuada P (2010) An ISS self-triggered implementation of linear controllers. Automatica 46(8):1310–1314
11. Mazo M, Tabuada P (2010) Input-to-state stability of self-triggered control systems. In: Proceedings of the, IEEE conference on decision and control, 2009 held jointly with the 2009, Chinese control conference. CDC/CCC 2009, vol 54, pp 928–933
12. Gommans TMP, Heemels WPMH (2015) Resource-aware MPC for constrained nonlinear systems: a self-triggered control approach. Syst Control Lett 79:59–67
13. Wang X, Lemmon MD (2009) Self-triggered feedback control systems with finite-gain stability. IEEE Trans Autom Control 54(3):452–467
14. Gommans T, Antunes D, Donkers T, Tabuada P, Heemels M (2014) Self-triggered linear quadratic control. Automatica 50(4):1279–1287
15. Santos C, Mazo M Jr, Espinosa F (2014) Adaptive self-triggered control of a remotely operated p 3-dx robot: simulation and experimentation. Rob Auton Syst 62(6):847–854

16. Poggio T, Girosi F (1990) Networks for approximation and learning. Proc IEEE 78(9): 1481–1497
17. Hornik K., Stinchcombe M., White H. (1989) Multilayer feedforward networks are universal approximators. Elsevier Science Ltd
18. Yi J, Wang Q, Zhao D, Wen JT (2007) Bp neural network prediction-based variable-period sampling approach for networked control systems. Appl Math Comput 185(2):976–988
19. Tu S (2011) Attitude dynamics and control of spacecraft. China Astronautic Publishing House, Beijing
20. Sun Z, Deng Z, Zhang Z (2011) Intelligent control theory and technology. Tsinghua University Press, Beijing