NEURAL NETWORK CONTROL OF VIETCOBOT

Chung Tan Lam[*] and Truong Trong Toai[+]
[*]Posts and Telecommunications Institute of Technology
[+] 3C Machinery Co., Ltd

Abstract: This paper is the research to come of our developing cobot, VietCobot, for the applications of service and medical. In this paper a neural network control is applied to a six degrees of freedom collaborative robot to observe the system dynamic behavior which plays an important role in force control. To do so, the dynamic equation of the cobot is derived taking into account parametric uncertainties and external disturbances. The forward kinematics and reverse dynamics of the cobot is solved; then, a neural network controller is derived to compensate the parametric uncertainties and disturbances based on the dynamic desired trajectory and sensory data of the joints. The simulation results illustrate the performance of the neural network control strategy. In addition, the all-in-one cobot integrated development environment software is developed to integrate the controller simulation and the experimental works for a complete research tool.

Keywords: cobot, neural network control.

I. INTRODUCTION

In almost cases of industrial robots, the torque-based dynamic model cannot be used directly because they are not functionally designed on the basis of torque/force control, but servo control [1]. In other words, actuators of those robots are equipped with servo motors that are controlled by input voltage, not by current. As the consequence, it is impossible to apply modern control method to them; and a new generation of robot is increasingly developed, that is cobot. Cobot can meet the needs of the torque-based control that may require in applications with collaboration, especially in service and medical fields.

Cobot has been increasingly enriched thesedays. Cobot has the advantages of a small structure, high reliability, small power consumption, easy to handle, and low price. These features make the cobot effective for broadly applying to many industries. Today, cobots are not only revolutionising the manufacturing industry, they are also being used in conjunction with current technologies to innovate service industries and to our lives [6].

Contact author: Chung Tan Lam
Email: lamct@ptithcm.edu.vn
Manuscript Received: 17/9/2021, revised: 17/10/2021, accepted: 20/10/2021.

For the controller design, most commercial industrial robot controllers implement some variety of PID control algorithm and modified mechanical structure which allow accuracy acceptable for many applications at a set of 'via' points specified by a human using the teach pendant; but it does not allow accurate dynamic trajectory following between the via points [1]. Moreover, the PID control accuracy is lost when unknown friction change, or for force control in surface finishing applications, and elsewhere. Adaptive control has proven successful in dealing with modeling uncertainties in general nonlinear systems by online tuning of parameters. A serious problem in using adaptive control on robotics is the assumption of linearity in the unknown system parameters; furthermore, this assumption requires one to determine the regression matrix for the system that can involve tedious computations, and a new regression matrix must be computed for each different robot [2]. Neural networks (NN for short), with their strong learning capability, have proven to be a suitable tool for controlling complex nonlinear dynamic system. The basic idea behind neural network control is to use a neural network estimator to identify the unknown nonlinear dynamics and compensate it. Also, the neural network-based approach can deal with the control of nonlinear system that may not be linearly parameterizable, that required in the adaptive control [1].

In this paper, a neural network controller, refer to [5], is considered for the joint-space position control and is applied to our developping cobot. The controller output is composed of a classical PID control and a neural network compensation term. The compensation term is used for online estimation of unknown nonlinear dynamics caused by parameter uncertainty and disturbances. No preliminary learning stage is required for the neural network weights. The controller is capable of disturbance-rejection in the presence of unknown bounded disturbances. Also, the implementation of hardware and software with realtime scheme is developed for the experiment in the next works.

II. DYNAMICS MODEL OF THE COBOT

In most applications cobot perform tasks which are depicted as certain desired trajectories. In order to achieve these tasks, joint-space reference trajectories $q_d(t)$ are denfined. Joint torque commands are then generated by the controller to make the cobot track the reference joint-space trajectories, and the desired task trajectories are tracked, respectively.

A. Forward Kinematics Invert Dynamics of the Cobot

The structure model of Vietcobot is shown in Fig. 1. The 7 frames $O_i X_1 Y_1 Z_1$ (i=0-6) are represented with parameter. θ_i, d_i, b_i, a_i, α_i represent the link rotation, link offset, shift, link length, and link twist respectively. The *Extended Denavit and Hartenberg* parameters of the cobot is given in Table I.

Initially, the algorithm calculates the absolute position, speed and acceleration of all the links of the robot. This task is iteratively executed to evaluate the kinematic quantities of the links, starting from the base of the cobot and proceeding to the end-effector. Conversely, the dynamic analysis is iteratively executed from the end-effector to the base.

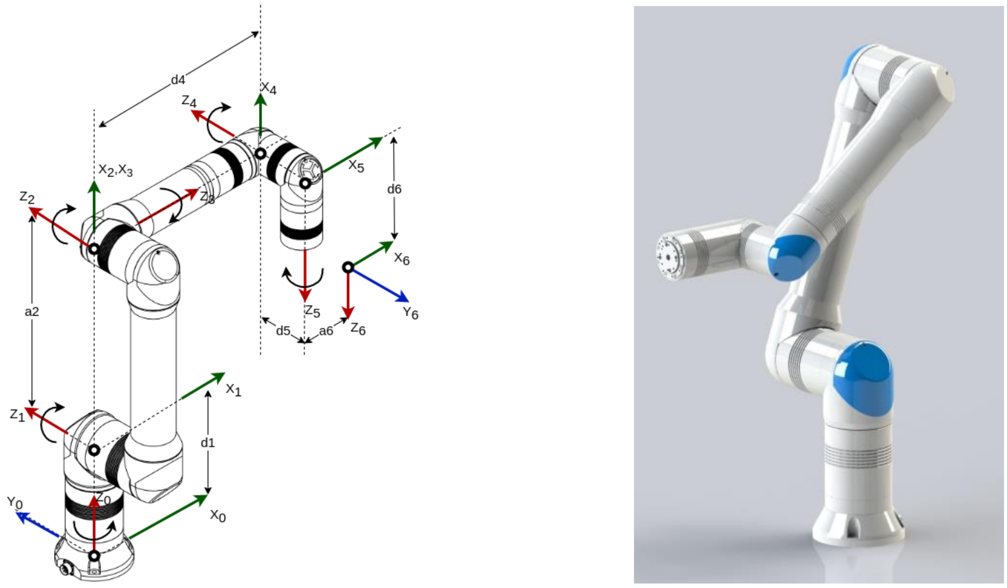


Figure 1. Structure of the Vietcobot and the 3D model

TABLE I. D-H Parameters

Frame	Theta (rad)	d_i (mm)	b_i (mm)	a_i (mm)	alpha (rad)
1	0	248	0	0	0
2	-Pi/2	0	0	460	-Pi/2
3	0	0	0	0	0
4	0	390	0	0	0
5	Pi	-110.5	0	0	Pi
6	0	170	0	0	0

The direct kinematics and the inverse dynamics for the cobot as following steps:

STEP 1: initialization procedures to change from the scalar to the matrix. Read the robot description, the number of links, joint type, five parameters to describe the position of the frame (i), fixed on link (i), with respect to the frame (i-1), fixed on link (i-1). The six barycentral inertial moments, the mass, the coordinates of the center of mass referred to the local frame (i), the three components of gravity acceleration referred to the base frame.

STEP 2: reads the joints motion.

STEP 3: for each link "i" and for each instant: the relative position, speed and acceleration of frame (i) with respect to frame (i-1).

STEP 4: the relative velocity and acceleration matrices by means **L** matrix

STEP 5: Evaluates the absolute position **M0** of each link (according to D-H. parameters)

$$M0_{0,i} = M0_{0,i-1} A_{i-1,i}$$

STEP 6-7: Transforms the relative velocity and acceleration matrices from local to the absolute frame (0)

$$W_{i-1,i(0)} = M0_{0,i-1} W_{i-1,i} M0_{0,i-1}^{-1}$$

$$H_{i-1,i(0)} = M0_{0,i-1} H_{i-1,i} M0_{0,i-1}^{-1}$$

STEP 8: Evaluates the absolute speed of each link by summing the drag and the relative speed of each link

$$W_{0,i} = W_{0,i-1} + W_{i-1,i(0)}$$

STEP 9: Evaluates of the absolute acceleration of each link by means the Coriolis' theorem

$$H_{0,i} = H_{0,i-1} + H_{i-1,i(0)} + 2W_{0,i-1} W_{i-1,i(0)}$$

STEP 10: External action

STEP 11: refer inertia matrix to absolute Frame (0)

STEP 12: evaluate inertia and weight action on link i

STEP 13: evaluate the constrain action on joint i

The numerical computation is shown in Fig 2.

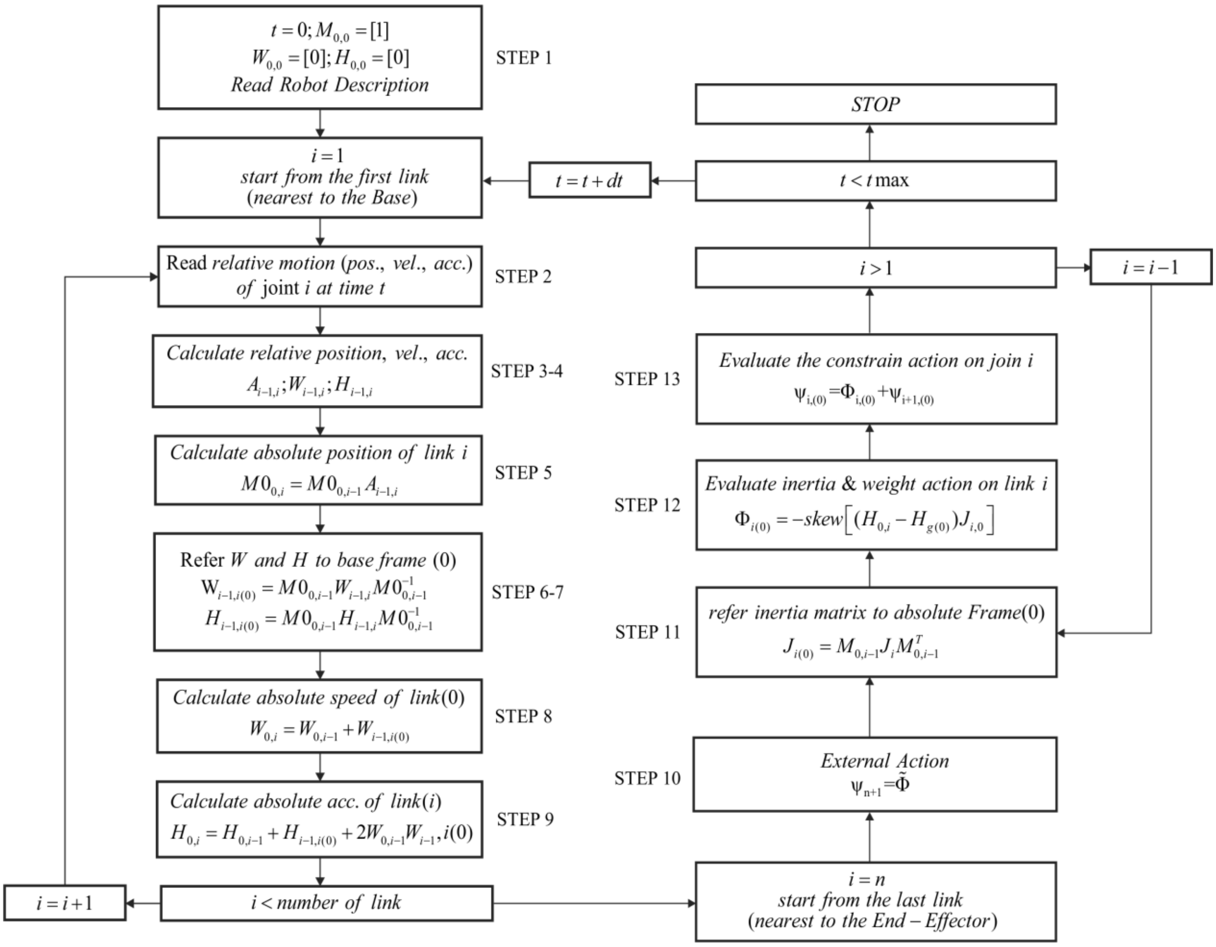


Figure 2. Flow chart for forward kinematics and invert dynamics

B. Dynamics Modeling of the Cobot

The cobot used in this paper is of six degrees of freedom configuration. The joints are controlled by integrated actuators, Fig. 18. The dynamics of the cobot subject to kinematics constraints is given in the following form [7]:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+F+\tau_d=\tau \tag{1}$$

where

$M(q)$: the inertia matrix

$C(q,\dot{q})$: Coriolis/centripetal matrix

q: generalized coordinate vector of the joint

$\dot{q},\ddot{q}$: velocity and acceleration of each joint

$F=g(q)+F(\dot{q})$: gravitational and friction force

τ_d: disturbance torque, τ: sum of all joint torques

In this equation, the unknown terms need to be compensated are the gravitational and friction force F and the disturbance τ_d on the robot.

Property 1: $\dot{M}(q)-2C(q,\dot{q})$ is skew-symmetric

Property 2: $M(q)$ symmetric, positive definite and bounded, $\|M(q)\|\le M_b$

Property 3: $C(q,\dot{q})\dot{q}$ is quadratic in $\dot{q}$, C is bounded, $\|C(q,\dot{q})\|\le C_b\|\dot{q}\|$

Assumption 1: Disturbances on the manipulator are bounded, that is, $\|\tau_d\|\le\tau_b$, with τ_b is a positive constant.

Assumption 2: Friction and gravity are bounded by $\|F(q,\dot{q})\|\le\xi_1+\xi_2\|\dot{q}\|$, where ξ_1 and ξ_2 representing some positive constant

C. Lyapunov Function of the Cobot

Let us define the tracking error and its derivative as

$$\begin{aligned} e&=q_d-q\\ \dot{e}&=\dot{q}_d-\dot{q}\end{aligned} \tag{2}$$

Also, the *filter tracking error* and its derivative,

$$\begin{aligned} r&=\dot{e}+ke, k=k^T>0\\ \dot{r}&=\ddot{e}+k\dot{e}=\ddot{q}_d-\ddot{q}+k(r-ke)\end{aligned} \tag{3}$$

From (2) and (3), we have

$$\begin{aligned} \dot{q}&=\dot{q}_d-(r-ke)\\ \ddot{q}&=\ddot{q}_d-\dot{r}+k(r-ke)\end{aligned}$$

The manipulator dynamics equation can be formulated in terms of filtered tracking error as follows:

$$M(\ddot{q}_d-\dot{r}+k\dot{e})+C(\dot{q}_d-r+ke)+F+\tau_d=\tau$$
$$M\dot{r}=-\tau+M(\ddot{q}_d+k\dot{e})+C(\dot{q}_d+ke)-Cr+f+\tau_d \quad (4)$$
$$M\dot{r}=-\tau+\left(Mk-C\right)(r-ke)+f+\tau_d$$

where $f=M\ddot{q}_d+C\dot{q}_d+F$

To design the manipulator torque input, the Lyapunov function for the manipulator is defined as

$$V=\frac{1}{2}r^TMr \quad (5)$$

the time derivative of V can be derived as follows

$$\begin{aligned}\dot{V}&=r^TM\dot{r}+\frac{1}{2}r^T\dot{M}r\\&=r^T\left\{M\dot{r}+\frac{1}{2}\dot{M}r\right\}\\&=r^T\left\{-\tau+Mkr-(Mk-C)ke+f+\tau_d\right\}+\frac{1}{2}r^T(\dot{M}-2C)r\\&\le r^T[-\tau+Mkr+(C-Mk)ke+f+\tau_d]\\&\le r^T\left\{-\tau+\psi\right\}+r^T\tau_d\end{aligned} \quad (6)$$

with the unknown nonlinear term

$$\psi=Mkr+(C-Mk)ke+M\ddot{q}_d+C\dot{q}_d+F \quad (7)$$

The nonlinear terms and in Eq. (7) are to be identified on-line using NN estimators. In the development of the NN on-line estimators, radial basis function (RBF) network with fixed centers and widths is employed. And it is not difficult to extend the results to the case where a multilayer perceptions (MLPs) network [3] is used.

III. NEURAL NETWORK CONTROLLER DESIGN

The RBF network has been shown to have universal approximation ability to approximate any smooth function on a compact set S, simply connected set of R^n [7], [20]. Let $f(.):S\to R^n$ be a smooth function and $\left\{\phi(x)\right\}$ be a basis set, where $f(.)\in C^m(S)$, the space of continuous functions. Then, for each $f(.)\in C^m(S)$, there exist a weight matrix W such that

$$f(x)=W^T\varphi(x)+\varepsilon \quad (8)$$

with the estimation error bounded by

$$\|\varepsilon\|<\varepsilon_N \quad (9)$$

for a given constant ε_N. It was shown in [20] that the set of RBFs forms a basis.

In light of the universal approximation ability of the RBF network ψ defined in Eq. (7), respectively, may be identified using RBF nets with sufficiently high number of hidden-layer neurons such that

$$\psi=W^Th(x)+\varepsilon(x) \quad (10)$$

where x is the input pattern to the neural networks defined as

$$x\equiv\left\{e^T \quad r^T \quad q_d^T \quad \dot{q}_d^T \quad \ddot{q}_d^T\right\}^T \quad (11)$$

$W\in R^{n_2xn}$ are ideal and unknown weights, respectively, which are assumed to be constant and bounded by

$$\|W\|_F\le W_B \quad (12)$$

with W_B some known positive constants; n_2 are the numbers of hidden-layer neurons of the RBF net, respectively; the approximation errors $\varepsilon\in R^n$ are bounded by $\|\varepsilon\|\le\varepsilon_N$, with ε_N positive constants; $h(x)$ are properly chosen RBFs for the hidden-layer neurons of the net, with x as the input pattern of the input layers to the net.

The basis functions can be chosen as the *Gaussian functions* defined as [7]

$$h_i(x)=\exp\left(\frac{-\|x-c_i\|^2}{\sigma_i^2}\right), i=1,2,..,n_2 \quad (13)$$

where c_i is center, and σ_i is width, which are all chosen a priori and kept fixed throughout for simplicity. Therefore, only the weights W is adjustable during the learning process. The estimates of ψ are given by

$$\hat{\psi}=\hat{W}^Th(x) \quad (14)$$

Thus, the main objective is to design proper control laws, that is, the torque inputs in Eq. (1), and stable NN learning laws, such that the unknown robot dynamics of Eq. (7) can be largely compensated for by the NN estimators, and the stability of the robot error dynamics in Eq. (4) and the boundedness on the NN estimation weights can be guaranteed. This is achieved through the following theorem.

Theorem: By choosing the control law for Eq. (1) as

$$\tau=kr+\hat{\psi} \quad (15)$$

where the weight updating law for the neural net as

$$\dot{\hat{W}}=\beta hr^T-\mu\beta\left\|r^T\right\|\hat{W} \quad (16)$$

where

$k>0$: control gain

β: positive constants representing the learning rate of the neural net

μ: small positive design parameter

By properly choosing the control gain and the design parameter, the tracking error of error dynamics described in Eq. (7), and the NN estimation weights W are all guaranteed to be *uniformly ultimately bounded*.

The unknown dynamics ψ is identified and compensated for in real time by the NN on-line estimator using the weight learning rule Eq. (16). In the NN estimator, the dynamic cobot is identified by using the common NN input pattern x which is defined in Eq. (11) and overall tracking error r^T.

To prove the theorem, we first introduce the following Lemma.

Lemma 1 (Bound on NN input x): It is assumed that the desired trajectory of the cobot is bounded as follows:

$$\left\| \begin{matrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{matrix} \right\| \le q_B \tag{17}$$

There exist computable positive constants q_B, c_0 and c_1, such that

$$\|x(t)\| \le c_1 + c_2 \|r\| \le q_B + c_0 \|r(0)\| + c_1 \|r(t)\| \tag{18}$$

Proof: the proof is straightforward following the proof of Lemma 4.1.1 in [22].

Proof of the theorem:

Assume that Eq. (10) holds, for all x in the compact set $S_x \equiv \left\{ x \mid \|x\| < b_x \right\}$, where $b_x > q_B$ - Let us define a compact set $S_r \equiv \left\{ r \mid \|r\| < (b_x - q_B)/(c_0 + c_1) \right\}$ with $r(0) \in S_r$ [2]. From inequality (18) we may show that the NN approximation property also holds for all P in the compact set S_r.

Substituting Eq. (15) into (6) yields

$$\begin{aligned} \dot{V} &\le r^T \left\{ -kr - \hat{\psi} + \psi \right\} + r^T \tau_d \\ &\le r^T \left\{ -kr - \hat{W}^T h + W^T h + \varepsilon \right\} + r^T \tau_d \\ &\le -kr^T r + r^T (\tilde{W}^T h) + r^T \varepsilon + r^T \tau_d \end{aligned} \tag{19}$$

where $\tilde{W} = W - \hat{W}$

Based on the bounds of every element of the vectors and matrices defined above $\varepsilon, \tau_d, W, h(x)$, we may show that the following properties hold:

$$\|\varepsilon\| \le \varepsilon_N, \|\tau_d\| \le \tau_N, \|W\|_F \le W_B \tag{20}$$

From (19) and (20) it follows that

$$\begin{aligned} \dot{V} &\le -kr^T r + r^T (\tilde{W}^T h) + r^T (\varepsilon + \tau_d) \\ &\le -k\|r\|^2 + r^T (\tilde{W}^T h) + \|r\| (\varepsilon_N + \tau_N) \end{aligned} \tag{21}$$

Let us choose the Lyapunov function as

$$V_1 = V + \frac{1}{2\beta} tr \left\{ \tilde{W}^T \tilde{W} \right\} \tag{22}$$

The Lyapunov function V_1 consists of the Lyapunov function V proposed in Eq. (5) for the cobot, and an additional term which are used to count for the NN learning dynamics.

Differentiating (22) and substituting (21) into it yields

$$\begin{aligned} \dot{V} &\le -k\|r\|^2 + r^T (\tilde{W}^T h) + \|r\| (\varepsilon_N + \tau_N) - \frac{1}{\beta} tr \left\{ \tilde{W}^T \dot{\hat{W}} \right\} \\ &= -k\|r\|^2 + \|r\| (\varepsilon_N + \tau_N) - \frac{1}{\beta} tr \left\{ \tilde{W}^T (\dot{\hat{W}} + \beta h r^T) \right\} \end{aligned} \tag{23}$$

Substituting Eq. (16) into (23) we obtain

$$\begin{aligned} \dot{V} &\le -\bar{k} \|r\|^2 + \|r\| (\varepsilon_N + \tau_N) + \mu \|r\| tr \left\{ \tilde{W}^T \hat{W} \right\} \\ &= -k\|r\|^2 + \|r\| (\varepsilon_N + \tau_N) + \mu \|r\| tr \left\{ \tilde{W}^T \hat{W} \right\} \end{aligned} \tag{24}$$

From the matrix theory, the following property holds [7]:

$$\begin{aligned} tr \left\{ \tilde{W}^T \hat{W} \right\} &= tr \left\{ \tilde{W}^T (W - \tilde{W}) \right\} \\ &= \left\langle \tilde{W}, W \right\rangle_F - \|\tilde{W}\|_F^2 \le \|\tilde{W}\|_F \|W\|_F - \|\tilde{W}\|_F^2 \end{aligned} \tag{25}$$

therefore

$$\begin{aligned} \dot{V} &\le -k\|r\|^2 + \mu \|r\| \|\tilde{W}\|_F - \mu \|P\| \|\tilde{W}\|_F^2 + \|r\| (\varepsilon_N + \tau_N) \\ &\le -k\|r\|^2 + \mu \|r\| \|\tilde{W}\|_F W_B - \mu \|r\| \|\tilde{W}\|_F^2 + \|r\| (\varepsilon_N + \tau_N) \\ &= -\|r\| \left\{ k\|r\| - \mu \|\tilde{W}\|_F W_B + \mu \|\tilde{W}\|_F^2 - (\varepsilon_N + \tau_N) \right\} \\ &= -\|r\| \left\{ k\|r\| + \mu \left(\|\tilde{W}\|_F - \frac{W_B}{2} \right)^2 - \left(\frac{\mu W_B^2}{4} + \varepsilon_N + \tau_N \right) \right\} \end{aligned} \tag{26}$$

which is guaranteed negative as long as

$$\|r\| > \frac{\frac{1}{4} \mu W_B^2 + \varepsilon_N + \tau_N}{k} \equiv b_r \tag{27}$$

$$\text{or} \quad \|\tilde{W}\|_F > \frac{W_B}{2} + \sqrt{\frac{\mu W_B^2}{4} + \frac{\varepsilon_N + \tau_N}{k}} \tag{28}$$

Furthermore, to ensure that the approximation property of the NN on-line estimators Eq. (14) holds throughout, the tracking error r should be always kept in the compact set S_r. This may be achieved by selecting the minimum control gain k to satisfy

$$k > \frac{\left(\frac{1}{4} \mu W_B^2 + \varepsilon_N + \tau_N \right) (c_0 + c_1)}{b_x - q_B} \tag{29}$$

Therefore, the compact set defined by $\|r\| \le b_r$ is contained within S_r; as a result, the approximation property of the NN estimator holds throughout. Thus, V_1 is negative outside a compact set. According to the standard Lyapunov theory and an extension of LaSalle theory [2], this demonstrates the uniform ultimate boundedness of the tracking error r, and the neural net weight $\tilde{W}$, and subsequently, the weight estimate $\hat{W}$ (noting that W is constant). Therefore, the control torques Eq. (15) are also guaranteed to be bounded. Moreover, the norm of the tracking errors $\|r\|$ can be kept arbitrarily small by increasing the minimum gain k in Eq. (27).

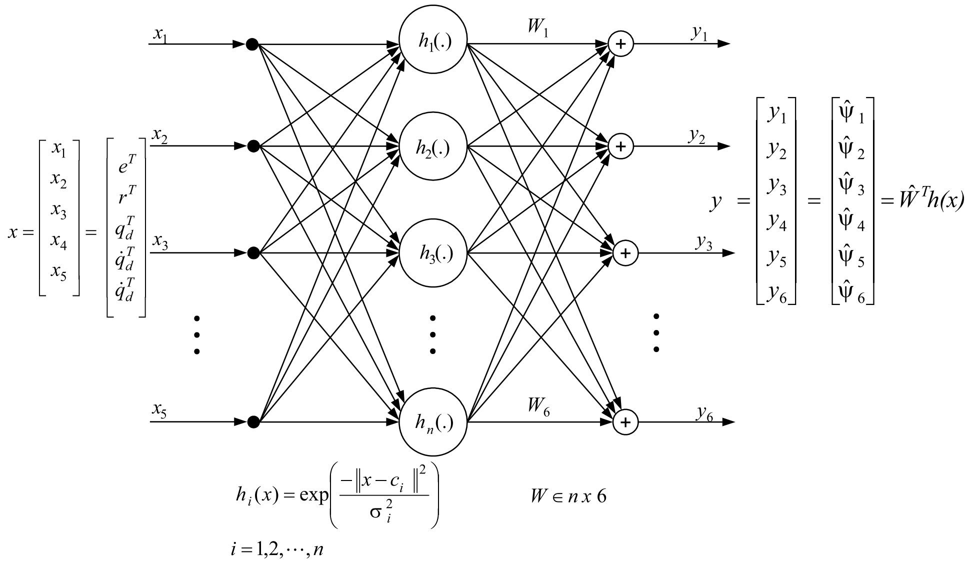


Figure 2. Neural network for the controller

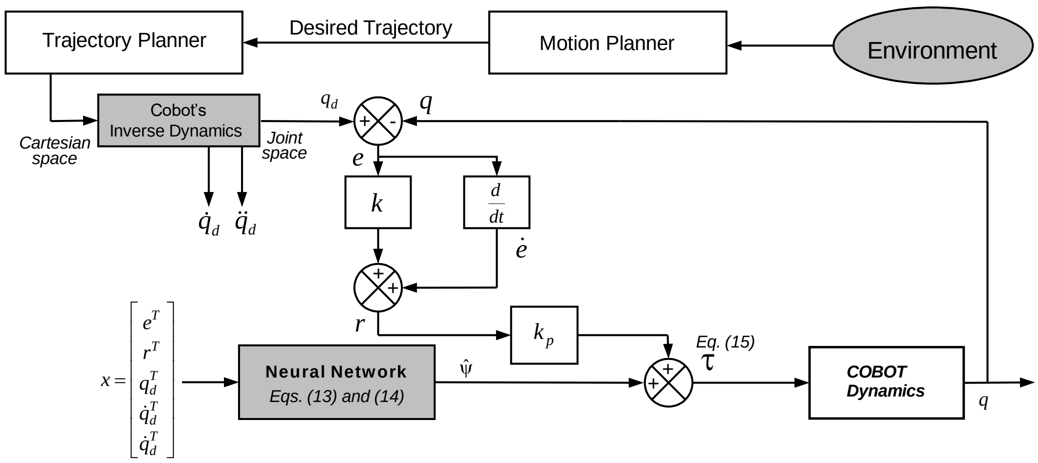


Figure 3. Neural network controller design

To verify the effectiveness of the controller, the simulations have been done with controller Eq. (4) using Cobot IDE in C#. The inversed kinematics of the cobot is skipped for the simple task of simulation. Instead, the desired trajectory is planned with sinusoid trajectory for joint 1 to joint 6 with the frequency of $0.8f, 0.9f, f, 1.1f, 1.2f$ and $1.3f$ ($f = 0.3125Hz$) as shown in Fig. 5. the initial joint positions, $Q(0) = (\pi/6, \pi/10, \pi/18, -\pi/18, -\pi/10, -\pi/6)$. The dynamics parameters of link 1 are extracted from CAD file using SolidWorks, refer to Figs. 4 and Table II. The parameters of the other links are not shown in Table II for the limited space of this writting. Sampling time is 0.01s. The tracking positions of joint 1 to joint 6 are shown in Figs. 6-11. The tracking errors of joint positions are shown in Figs. 12 and 13, it is shown that the performance of the controller is good enough. The estimated value for the manipulator $\hat{\psi}$ is given in Figs. 14 and 15. The output of the Gaussian function for the controller is given in Fig. 16. The joint torque plot is shown in Fig. 17.

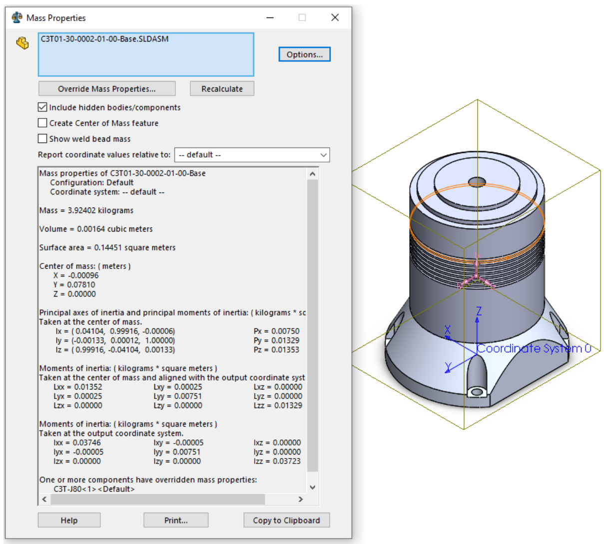


Figure 4. Dynamics paremeter of link 1

TABLE II. The cobot's dynamics parameters

Parameters	**VietCobot_Data.csv**
Number of Link	Number_of_Links,6
Link 1	[1].LINK_1
Joint Type	JointType,0
DH parameters	DH-Theta/d/b/a/alpha,0,0.248,0,0,-1.57079
Mass of the first link [kg]	Mass, 3.92402
Inertia moments jxx, jxy, jxz	Jxx/Jxy/Jxz,0.03746,-0.00005,0.0
Inertia moments jyy, jyz	Jyy/Jyz,0.00751,0.0
Inertia moments jzz [kg.m^2]	Jzz,0.03723
CoM Xg, Yg, Zg [m]	CoM-Xg/Yg/Zg,-0.00096,0.07810,0.0
Link 2	[2].LINK_2
…	…
Link 6	[6].LINK_6
…	…
External Action	
Gravity in BASE frame (0)	Gravity,0,0,-9.8
External forces and torques on End-effector	Ext_Force,0,0,0,0,0,0

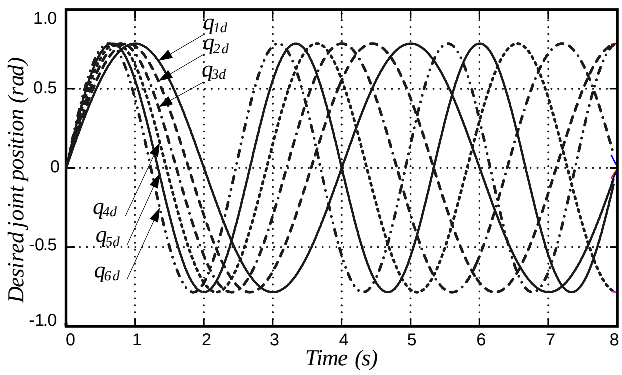


Figure 5. Joint desired trajectory

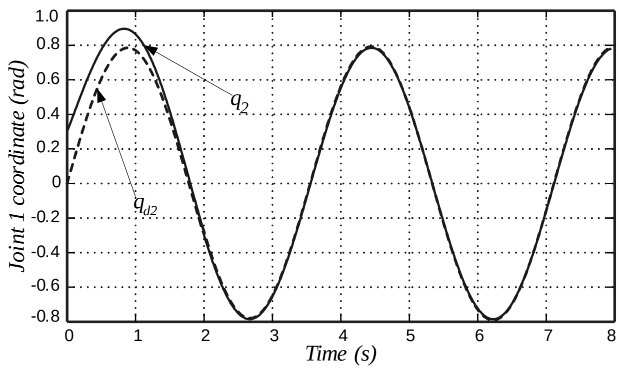


Figure 7. Tracking position of joint 2

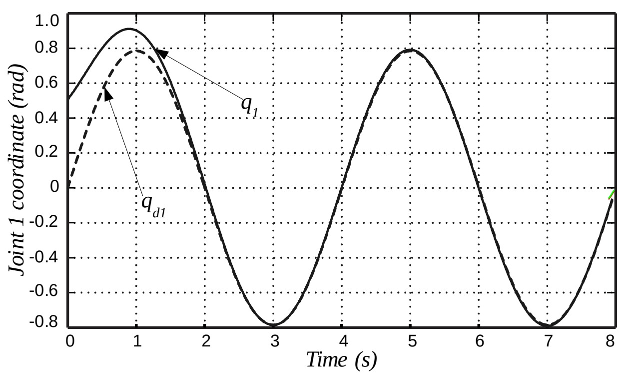


Figure 6. Tracking position of joint 1

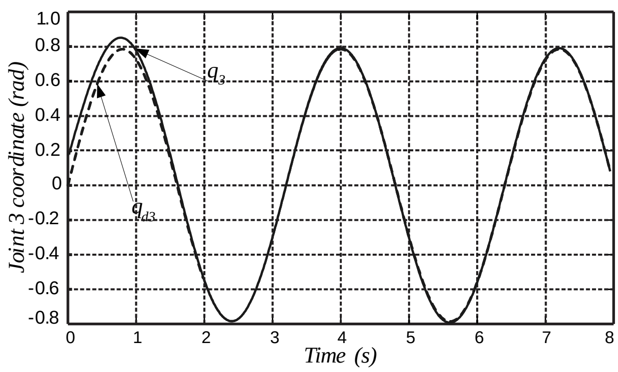


Figure 8. Tracking position of joint 3

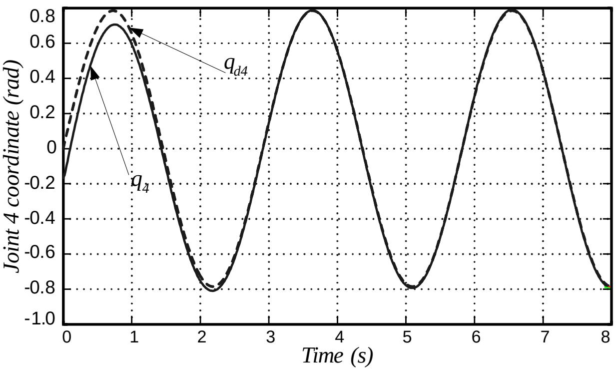


Figure 9. Tracking position of joint 4

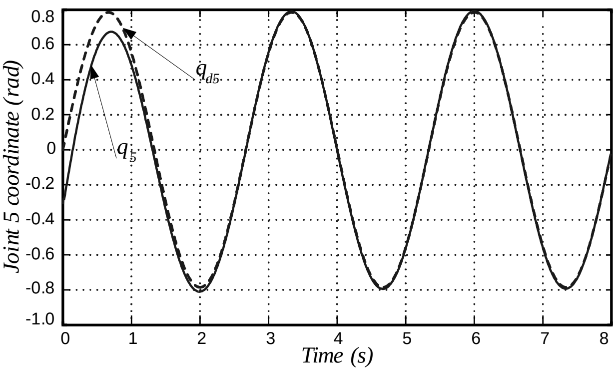


Figure 10. Tracking position of joint 5

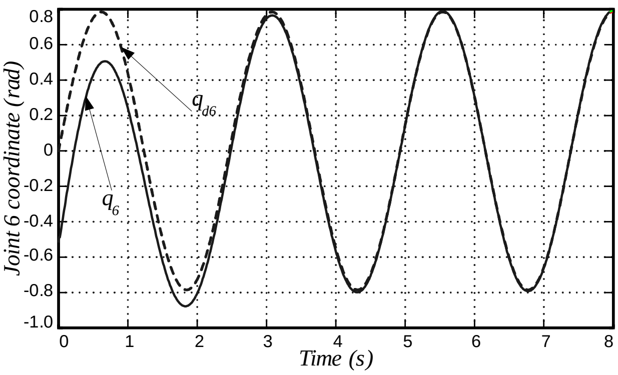


Figure 11. Tracking position of joint 6

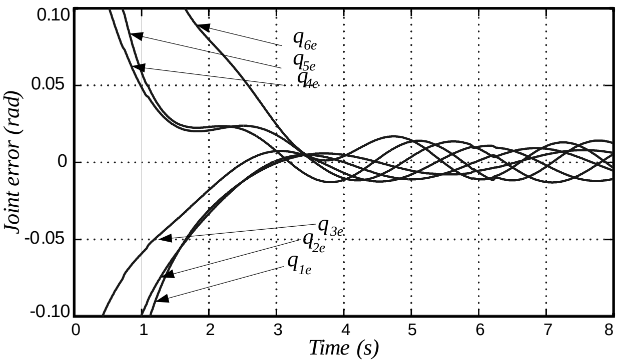


Figure 12. Joint position error

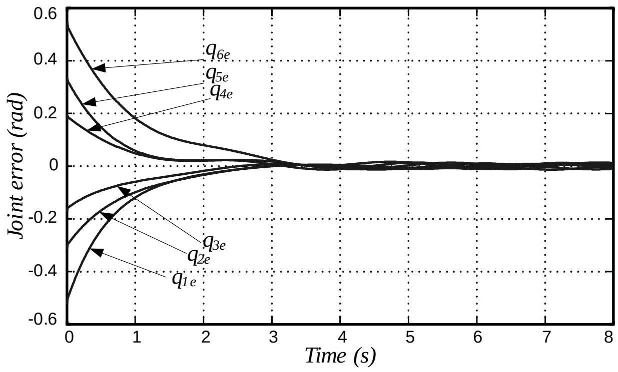


Figure 13. Joint position error

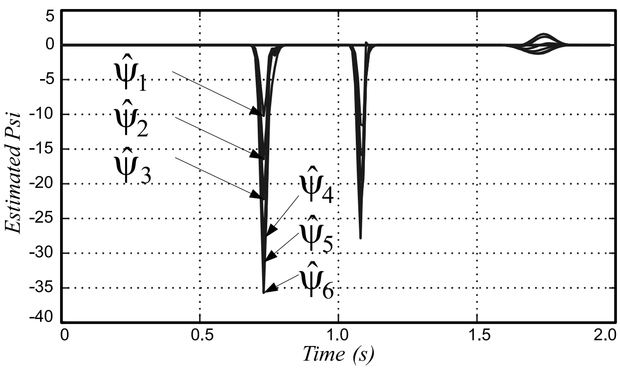


Figure 14. Estimated $\hat{\psi}$ for 0-2s

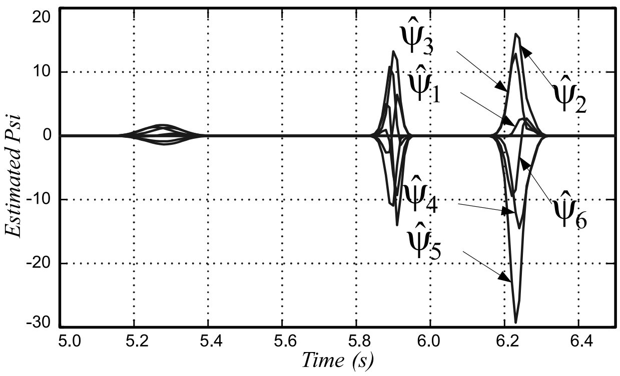


Figure 15. Estimated $\hat{\psi}$ for 5-6.5s

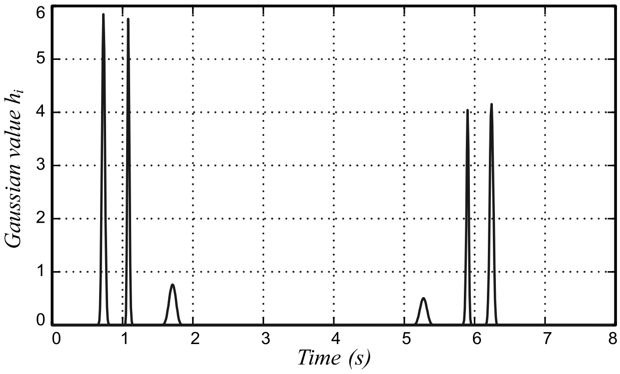


Figure 16. Gaussian value of hidden layer function

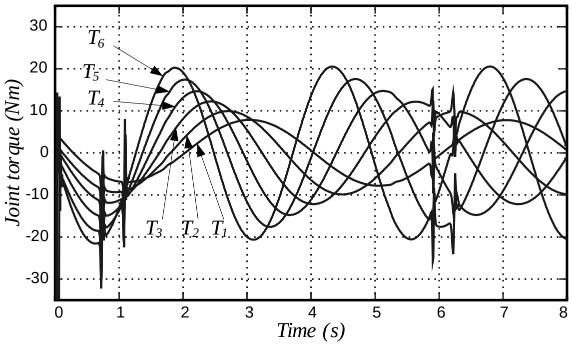


Figure 17. Joint torques τ

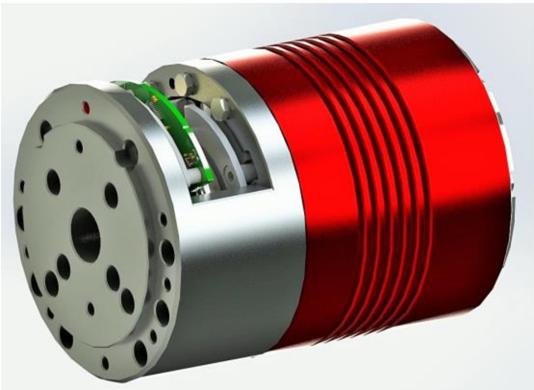
IV. CONTROL SYSTEM IMPLEMENTATION

This paper presents the next step of the development of Vietcobot in [12]. The cobot structure is modified for service and medical applications as well. The joint actuator is designed which is a combination of several componets: harmonic gearbox, motor, encoder, brack, servo driver, communications. The intergrated actuator which is used to control cobot joints is shown in Fig. 18

The control system is based on the PC-based control with realtime EtherCAT communications which sends the servo frames to the actuators of the cobot. There are 6 actuators are used to control 6 joints, respectively. The control input torques of the NN controller are transmitted to the actuators via EtherCAT communications in realtime to track a desired trajectory of the task. An Ultra-compact Industrial PC from Beckhoff with TwinCAT3 is used for the main processing. The overall control system is shown in Fig. 19. The total control are programmed in the Cobot IDE V.10 in C# with ADS of TwinCAT3 shown in Fig. 20.

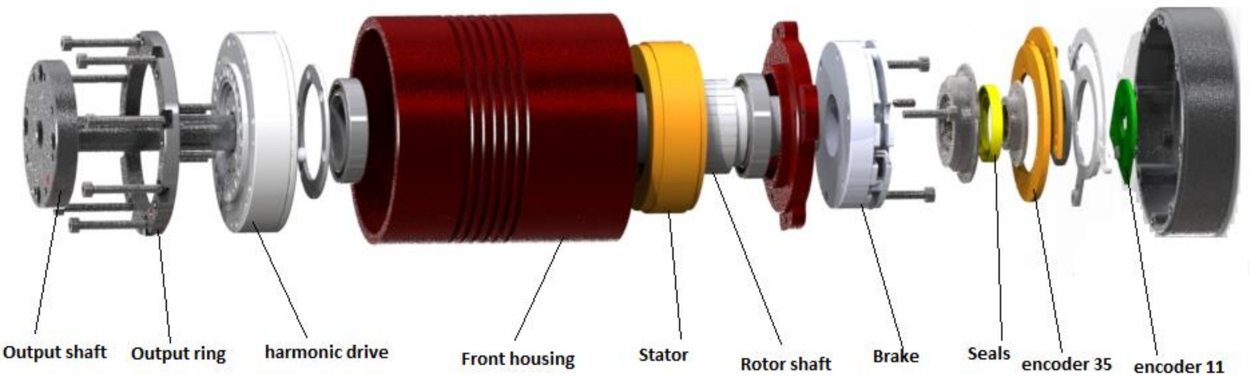
Vision of 2D/3D plays an important role in cobots; for example, the cobot can detect 2D/3D objects in the assembly lines for the picking tasks in factories without workers. At first, Vietcobot IDE integrated LotusVision tool and Intel RealSense camera to make it possible for 2D vision in some simple planar task only. For the realtime 3D Vision scheme, the LotusVision can be implemented in another industrial PC and the results of the processing is send back to main computer effectively.

A setup of cobot and the 3DVision can be applied to medical fields which needs to detect temperatures and positions on patients' body for a special curing method. This topic will be presented in our researches to come.



(1). Design of the actuator

(2) Prototype of the actuator (VietCobot)



(3) Assembling of the actuator

Figure 18. Integrated actuator for VietCobot

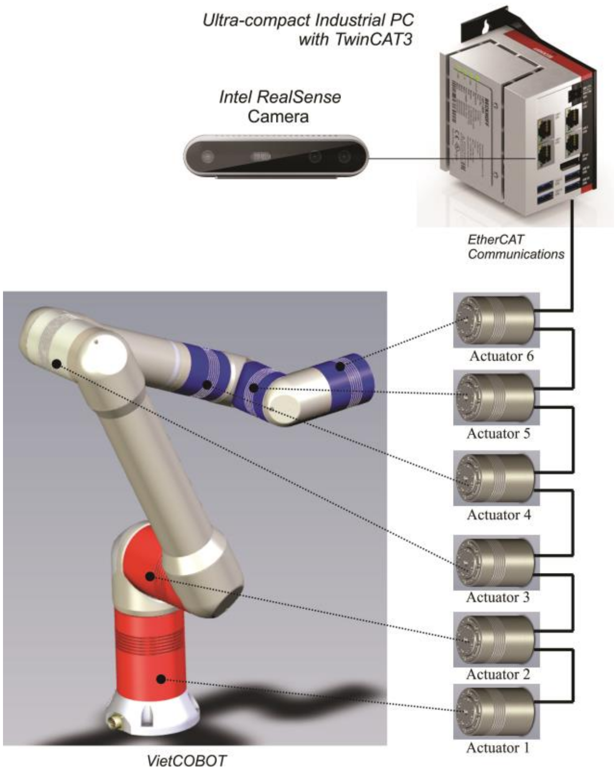


Figure 19. The hardware of the control system

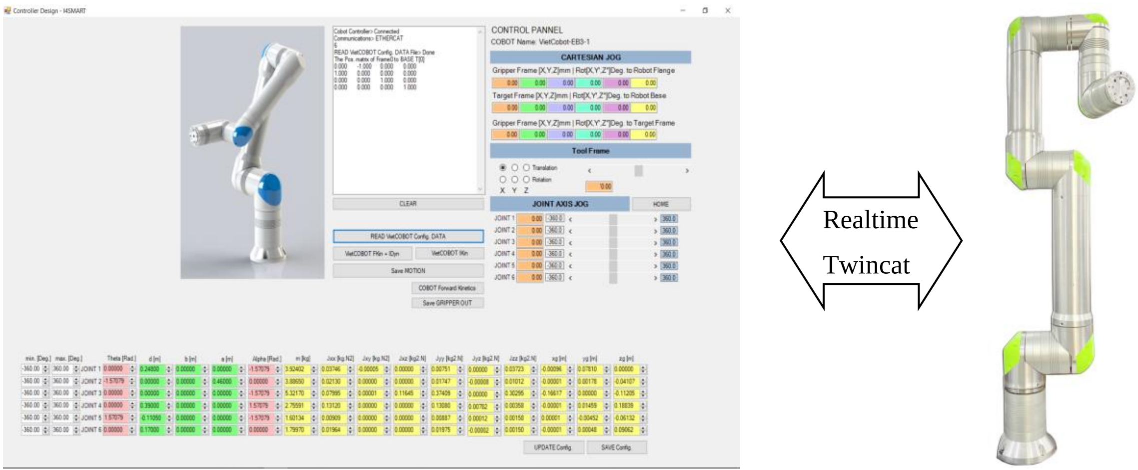


Figure 20. The Cobot IDE for simulation with realtime control

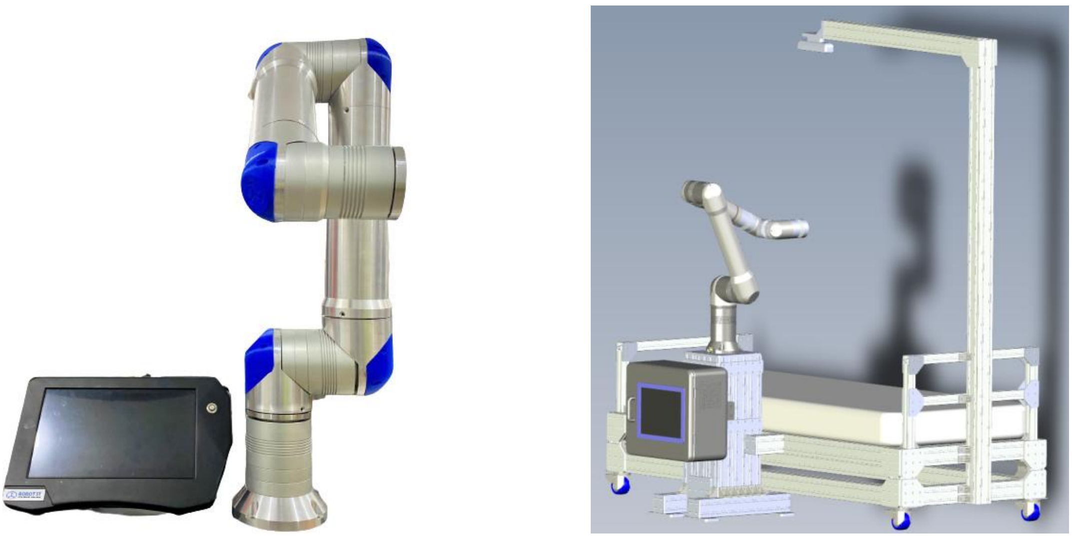


Figure 21. VietCobot in testbed and a proposed application in medical field (future works)

V. CONCLUSIONS

In this paper, the structure model of the cobot is developed for service and medical applications. The extended Denavit and Hartenberg parameters is designed for deriving the kinematics and invert dynamics of the cobot. The physical data, motion data, data structures, kinematics algorithm, dynamics controller development are defined for numerical computing on C# with ease. And, a neural network controller is designed for online estimation of unknown nonlinear dynamics caused by parameter uncertainty and disturbances. The control system and software tool are developed for torque-based control that is required the torque-based dynamics model in realtime, especially in service and medical applications. The experiment has not finished at the time of this writing, and it will be performed in the future works.

ACKNOWLEDGMENT

This work was funded by Vingroup Joint Stock Company and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.NCUD.DA059

REFERENCES

[1] F. L. Lewis, S. lagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems. London*, U.K.: Taylor and Francis, 1999.

[2] Alexander L. Fradkov, Iliya V. Miroshnik and Vladimir O. Nikiforov, *Nonlinear and Adaptive Control of Complex Systems*, Kluwer Academic Publishers, 1999

[3] Dan Zhang and Bin Wei, *Adaptive Control for Robotic Manipulators*: Taylor & Francis, 2017

[4] Ignacy Duleba, ”Structural Properties of Inertia Matrix and Gravity Vector of Dynamics of Rigid Manipulators,” Wiley Periodicals, Inc , Journal of Robotic Systems 19(11), 2002, pp. 555–567.

[5] S. Lin and A. A. Goldenberg, “Neural-Network Control of Mobile Manipulator,” IEEE Transactions on Neural Networks, Vol. 12, No. 5, September, 2001, pp. 1121-1133.

[6] Saixuan Chen, Minzhou Luo, “The Development of a New 6 DOF Collaborative Robot,” IEEE 9th International Conference on Mechanical and Intelligent Manufacturing Technologies, 2018, 165-171

[7] Ignacy Duleba, “Structural Properties of Inertia Matrix and Gravity Vector of Dynamics of Rigid Manipulators.” Wiley Periodicals, Inc, 2002.

[8] Xinjun Liu, Jingjuu, Guobiao Wang, “*Research trend and scientific challenge of robotics*,” China Academic Journal Electronic Publishing House. 2016, 5, pp. 425-431.

[9] Hai Wang, Banchen Fu, Bin Xue, “Dynamic analysis of 6-DOF flexible manipulator,” China Mechanical Engineering, 2016, 27(8), pp.1096-1101

[10] TwinCAT, [Online]. Available: https://www.beckhoff.com/

[11] Longzheng Tie, “Collaboration robots in Industrial 4.0,” China Electrical Equipment Industry, 2015(8), pp. 62-63

[12] Chung Tan Lam and Truong Trong Toai, “Development of a Collaborative Robot – VietCobot,” Journal of Science and Technology on Information and Communications,” CS.01 2021

ĐIỀU KHIỂN MẠNG NƠRÔN CHO ROBOT CỘNG TÁC

Tóm tắt: Bài báo này trình bày ứng dụng điều khiển mạng Nơ-rôn cho robot cộng tác 6 bậc tự do để quan sát hành vi hệ động lực học để ứng dụng trong các lĩnh vực dịch vụ và y tế. Chúng ta tiến hành xây dựng phương trình động lực học của cobot có đưa vào các tham số không chắc chắn và nhiễu tác động từ bên ngoài. Và thiết một bộ điều khiển mạng nơ-rôn để bù trừ nhiễu và các tham số không chắc chắn đó dựa trên quỹ đạo mong muốn của góc khớp và dữ liệu đưa về từ cảm biến của các khớp cobot. Các kết quả mô phỏng cho thấy hiệu quả của chiến lược điều khiển mạng nơ-rôn nêu trên. Ngoài ra, một phần mềm môi trường phát triển tích hợp cho cobot được phát triển tích hợp tính năng mô phỏng bộ điều khiển và thực hiện thực nghiệm đáp ứng yêu cầu một công cụ nghiên cứu đầy đủ.

Từ khóa: Robot, điều khiển mạng nơrôn.

Chung Tan Lam, He received Master of Mechanical Design from Department of Mechanical Engineering, Pukyong National University, Pusan, Korea in 2002. He received Ph.D of Mechatronics from Pukyong National University in 2006. Currently, he is Head of Automatic Control Division, Electronics Department 2, PTIT-HCM. His research interests are Factory Automation, Modelling and Control of Robotics, Embedded systems, Realtime communications in robotics, Cobots and Smart devices, Mobile robots.

Truong Trong Toai, He is Vice Director of 3C Machinery Co. Ltd., established in 2014. The company is one of the leading R/D in robotics fields in HoChiMinh City, Vietnam. The company got the Price“ Vietnam’s Rice Bowl Starup Awards” in 2017. His research interests are Solutions of Industrial Robotics for SMEs, Smart Devices, Modernizing Industrial Production Process, Collaborative robots and Smart Actuators.