Position Control of a PM Stepper Motor Using Neural Networks

Gang Feng
School of Electrical Engineering, University of New South Wales
Sydney, NSW 2052, Australia
Dept. of MEEM, City University of Hong Kong
Tat Chee Ave., Kowloon, Hong Kong
Email: megfeng@cityu.edu.hk

Abstract

This paper considers position control of a PM stepper motor. A new control scheme is proposed based on a kind of exact linearization controller and a neural network based compensating controller. This scheme takes advantages of simplicity of the model based control approach and uses the neural network controller to compensate for the motor modeling uncertainties. The neural network is trained on line based on Lyapunov theory and thus its convergence is guaranteed.

1. Introduction

Due to the various disadvantages of DC motors, positioning systems are more and more implemented by using induction motors or stepper motors. Originally, stepper motors were designed to provide precise positioning control within an integer number of steps without using any sensors. This is an open-loop operation. However, using the stepper motors in an open-loop configuration results in very low performance. In particular, the PM stepper motors have a step response with significant overshoot and a long settling time. Therefore, feedback control has been proposed for stepper motor position systems. A number of feedback control techniques have been developed during the last few years. One of them is the so-called feedback linearization controller [1-3]. The basic idea is to design a feedback controller so that the nonlinear stepper motor system becomes a linear system. Then the techniques of linear control systems can be used to design the controller for the linearized stepper motor system so that the required performance can be achieved.

It is understandable that such a control technique is capable of providing excellent positioning results if the complete dynamics of the stepper motor are known. However it actually results in no better performance than that of the conventional fixed gain controllers due to the fact that it is very hard to obtain perfect dynamic model for the stepper motors in practice. Moreover, there also exist uncertainties, as well as time-varying effects in practice. All these factors lead to the difficulty of achieving higher precision positioning performance.

Recently, increasing attention has been paid to the use of artificial neural networks in nonlinear control [4-6]. One possible way to use the neural networks is to replace either the entire control system or the feedforward controller with neural networks. Such research work was based on the desire to obtain the benefits of model-based control without *a priori* knowledge of system dynamics. However, in many cases, the nominal dynamic model of the stepper motor can be found *a priori* indeed. Therefore, *a priori* knowledge of stepper motor dynamic model should be appropriately used rather than totally discarded.

In this paper, a new scheme for stepper motor positioning control is proposed. This scheme takes advantage of simplicity of the conventional methods, and incorporates a compensating controller to achieve high positioning performance. The compensating controller is based on an RBF neural network, which is trained on-line to identify the stepper motor modeling uncertainties.

The rest of the paper is organized as follows. Section 2 presents the problem formulation and Section 3 proposes a new control scheme which is followed by some concluding remarks in section 4.

2. Problem formulation

The equations describing the stepper motor can be given as below [7],

$$
\begin{aligned}
\frac{di_a}{dt} &= [v_a - Ri_a + K_m \omega \sin(N_r\theta)]/L \\
\frac{di_b}{dt} &= [v_b - Ri_b - K_m \omega \cos(N_r\theta)]/L \qquad (1)\\
\frac{d\omega}{dt} &= [-K_m i_a \sin(N_r\theta) + K_m i_b \cos(N_r\theta) - G\omega \\
&\quad - K_D \sin(4N_r\theta)]/J - \tau_L / J \\
\frac{d\theta}{dt} &= \omega
\end{aligned}
$$

where $i_{a,} i_b$ and $v_{a,} v_b$ are the currents and voltages in phase A and B respectively, L and R are the self-inductance and resistance of each phase winding, K_m is the motor torque constant, N_r is the number of rotor teeth, J is the rotor inertia, G is the viscous friction constant, ω is the rotor speed, θ is the motor position, τ_L is the load torque, and the term $K_D \sin(4N_r\theta)$ represents the detent torque due to the permanent rotor magnet interacting with the magnetic material of the stator poles.

As shown in [1], if the modeling is perfect, with an appropriate nonlinear coordinate transformation and exact feedback linearization, the stepper motor equations can be rewritten as,

$$\dot{x} = Ax + Bu \tag{2}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$[x_1 \quad x_2 \quad x_3] := [\theta \quad \omega \quad \alpha] / K_3,$$
$$\alpha := \dot{\omega}, \; K_3 := K_m / J,$$

and x_4 is related with the maximum rated phase current. Then a possible linear control law for u can be chosen as,

$$u = -\begin{bmatrix} a_{30} & a_{31} & a_{32} & 0 \\ 0 & 0 & 0 & a_{40} \end{bmatrix} \begin{bmatrix} x_1 - x_{1d} \\ x_2 - x_{2d} \\ x_3 - x_{3d} \\ x_4 \end{bmatrix} + \begin{bmatrix} r(t) \\ 0 \end{bmatrix} \tag{3}$$

where $[x_{1d} \; x_{2d} \; x_{3d}] := [\theta_d \; \omega_d \; \alpha_d] / K_3$, θ_d, ω_d, α_d are the desired position, speed and acceleration profiles respectively, and $r(t) = \dot{\alpha}_d / K_3$. It can be seen that with the above controller, the closed loop system can be expressed as,

$$\dot{\tilde{x}} = A_c \, \tilde{x} \tag{4}$$

where

$$\tilde{x} := [x_1 - x_{1d} \quad x_2 - x_{2d} \quad x_3 - x_{3d} \quad x_4]^T$$

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_{30} & a_{31} & a_{32} & 0 \\ 0 & 0 & 0 & a_{40} \end{bmatrix}.$$

Therefore, the error dynamics of the positioning control system can be well designed by suitably choosing the coefficients a_{30}, a_{31}, a_{32} and a_{40}.

However, if the modeling is not perfect or there exist uncertainties which is always the case in practice, then the above design might lead to the significant degrade of the performance. In this case, we propose a new control scheme which includes a compensating controller. It is supposed that the nominal model of the stepper motor is known *a priori*. Then the feedback linearization control discussed above can be designed based on this nominal model. However, in this case, the stepper motor control system equation will not be expressed as in eqn.(2). Instead, there will exist an uncertainty term *f(x)* in the equation, which will be the unknown nonlinear function of the motor state variables. It is supposed that the uncertainty term satisfies the matching condition, that is, the term is in the range of the control signal *u(t)*. In such a case, the system can be expressed as,

$$\dot{x} = Ax + B[u + f(x)] \tag{5}$$

It can be clearly seen that the performance of the stepper motor position control system will degrade due to this uncertainty term *f(x)*. Our objective is to design a compensating controller to improve the stepper motor positioning performance. If the nonlinear function *f(x)* were known *a priori*, then a modified controller

$$u = u_o + u_c \tag{6}$$

$$u_o = -\begin{bmatrix} a_{30} & a_{31} & a_{32} & 0 \\ 0 & 0 & 0 & a_{40} \end{bmatrix} \begin{bmatrix} x_1 - x_{1d} \\ x_2 - x_{2d} \\ x_3 - x_{3d} \\ x_4 \end{bmatrix} + \begin{bmatrix} r(t) \\ 0 \end{bmatrix} \tag{7}$$

$$u_c = -f(x) \tag{8}$$

would lead to the closed loop system expressed as in eqn.(4), i.e., the known modeling uncertainties could be well compensated.

Unfortunately, the non-linear function *f(x)* is unknown *a priori* in practice. Therefore the above modified controller could not be implemented. However, this controller suggests indeed that a well estimated function $\hat{f}(x)$ of the non-linear function *f(x)* could be used to improve the stepper motor positioning performance. It is noted that there exist another nonlinear function $h(\tilde{x})$ such that $f(x) = h(\tilde{x})$ by taking notice of the definition of the state variable $\tilde{x}$. With the same control law (6), (7) and a new compensating control law, which will be discussed subsequently, the closed loop stepper motor control system can be expressed as,

$$\dot{\tilde{x}} = A_c \, \tilde{x} + B[u_c + h(\tilde{x})] \tag{9}$$

Due to their great approximation capability, artificial neural networks will be used in this paper to identify this non-linear function [8-10]. For this we make the following assumptions.

Assumption 1: The closed loop stepper motor control system, whose controller is designed based on the nominal stepper motor model, is stable, and the tracking error vector $\tilde{x}$ belongs to a compact set.

Assumption 2:

(i) Given a positive constant ε_0 and a continuous function $h: C \to R$, where $C \subset R^{m_r}$ is a compact set, there exists a weight vector $\theta = \theta^*$ such that the output $\hat{h}(\tilde{x}, \theta)$ of the neural network architecture with n^* nodes satisfies

$$\max_{\tilde{x} \in C} | \hat{h}(\tilde{x}, \theta^*) - h(\tilde{x}) | \le \varepsilon_0 \,,$$

where n^* may depend on precision parameter ε_0 and the function *h*.

(ii) The output $\hat{h}(\tilde{x},\theta)$ of the neural network architecture is continuous with respect to its arguments for all finite ($\tilde{x}$, θ).

Remark 1: This paper is mainly concerned with the compensating controller design for positioning performance, thus it is reasonable to assume the control system based on the nominal model is stable in the assumption 1. Assumption 2 is also reasonable due to the universal function approximation capability of the neural networks.

3. A new control scheme

Suppose the unknown nonlinear function $h(\tilde{x})$ be parameterized by a static RBF neural network with output $\hat{h}(\tilde{x},\theta)$, where $\theta \in R^{n^*}$ is the adjustable weight, and n^* denotes the number of weights in the neural network approximation. Then the eqn.(2) can be rewritten as

$$\dot{\tilde{x}} = A_c\ \tilde{x} + B[u_c + \hat{h}(\tilde{x},\theta^*) + (h(\tilde{x}) - \hat{h}(\tilde{x},\theta^*))] \qquad (10)$$

where θ^* denotes the optimal weight values in the approximation for $\tilde{x}$ belonging to a compact set $C(M_x) \subset R^{2n}$, that is $C(M_x) := \{\tilde{x} : \|\tilde{x}\| \le M_x\}$. In general, the "optimal" weight θ^* in eqn.(10) could take arbitrarily large values. However, in order to avoid any numerical problems that may arise due to too large weights and to prevent the weights from drifting to infinity, we are only concerned with weights that belong to a large compact set $B(M_\theta)$, where M_θ is a design constant, and $B(M_\theta) := \{\theta : \|\theta\| \le M_\theta\}$ denotes a ball of radius M_θ. In the design of adaptive law, we also restrict the estimate of θ^* to the compact set $B(M_\theta)$ through the use of a projection approach. In this way, the optimal weight θ^* is defined as the element in $B(M_\theta)$ that minimizes the function $\| h(\tilde{x}) - \hat{h}(\tilde{x},\theta) \|$ for $\tilde{x} \in C$, i.e.

$$\theta^* := \arg \min_{\theta \in B(M_\theta)} \{ \sup_{\tilde{x} \in C(M_x)} \| h(\tilde{x}) - \hat{h}(\tilde{x},\theta) \| \}$$

Now eqn.(10) can be expressed as

$$\dot{\tilde{x}} = A_c\ \tilde{x} + B[u_c + \hat{h}(\tilde{x},\theta^*) + \eta(\tilde{x})] \qquad (11)$$

where η denotes the error due to the use of the neural network,

$$\eta(\tilde{x}) := h(\tilde{x}) - \hat{h}(\tilde{x},\theta^*) \qquad (12)$$

The error η is bounded by a finite constant vector η_0, where

$$\eta_0 := \sup_{\tilde{x} \in C} \| h(\tilde{x}) - \hat{h}(\tilde{x},\theta^*) \|$$

According to the properties of the RBF neural networks, the function $\hat{h}(\tilde{x},\theta^*)$ can be expressed in the form

$$\hat{h}(\tilde{x},\theta^*) = \theta^{*T} \phi(\tilde{x}) \qquad (13)$$

where θ^* is a matrix representing the optimal weight values subject to the constraints $\|\theta^*\| \le M_\theta$, the vector field $\phi(\tilde{x}) \in R^{n^*}$ which is refereed to regressor, is Gaussian type of functions defined element-wise as

$$\phi\ (\tilde{x}) = \exp(-\frac{|\tilde{x} - c_i|^2}{\sigma_i^2}), \qquad i = 1, 2, .., n^*.$$

For the sake of tractable analysis, c_i and σ_i are chosen *a priori* and kept fixed. The local training techniques presented in [11] could be used for appropriately choosing the centres and widths of the RBF neural network. Or the centres can be chosen as the mesh points equally distributed inside the compact set. In such case, the only adjustable parameters θ appear linearly with respect to the known nonlinearity $\phi(x)$.

Now, eqn.(11) can be rewritten as

$$\dot{\tilde{x}} = A_c\ \tilde{x} + B[u_c + \theta^{*T} \phi(\tilde{x}) + \eta(\tilde{x})] \qquad (14)$$

Next, we will show that an adaptive compensating controller can be added to achieve improved positioning performance. The new compensating control law is as follows:

$$u_c = -\hat{\theta}^T \phi(\tilde{x}) \qquad (15)$$

where $\hat{\theta}$ is the estimated parameter for θ^*.

This control law leads to the closed loop system expressed as

$$\begin{aligned}\dot{\tilde{x}} &= A_c\ \tilde{x} + B[\theta^{*T} \phi(\tilde{x}) + \eta(\tilde{x}) - \hat{\theta}^T \phi(\tilde{x})] \\ &= A_c\ \tilde{x} + B[-\tilde{\theta}^T \phi(\tilde{x}) + \eta(\tilde{x})] \qquad (16)\end{aligned}$$

where $\tilde{\theta} := \hat{\theta} - \theta^*$ denotes the parameter estimation error.

Choosing a Lyapunov function candidate

$$V = \frac{1}{2} \tilde{x}^T P \tilde{x} + \frac{1}{2\gamma} \| \tilde{\theta} \|_F^2 \qquad \gamma > 0 \qquad (17)$$

where $\| \cdot \|_F^2$ denotes the Frobenius matrix norm, defined as $\| R \|_F^2 := \sum_{ij} | r_{ij} |^2$, and the matrix *P* is positive definite and satisfies the following Lyapunov equation

$$PA_c + A_c^T P = -Q \qquad (18)$$

with $Q \ge 0$.

Taking the time derivative of *V* along the trajectories of eqn.(16), we have

$$\begin{aligned}\dot{V} &= \frac{1}{2}[\tilde{x}^T P \dot{\tilde{x}} + \dot{\tilde{x}}^T P \tilde{x}] + \frac{1}{\gamma} tr(\dot{\tilde{\theta}}^T \tilde{\theta}) \\ &= \frac{1}{2}[\tilde{x}^T (PA_c + A_c^T P)\tilde{x}] + [-\tilde{\theta}^T \phi(\tilde{x}) + \eta]^T B^T P \tilde{x} + \frac{1}{\gamma} tr(\dot{\hat{\theta}}^T \tilde{\theta}) \\ &= \frac{1}{2}[-\tilde{x}^T Q \tilde{x}] - \phi(\tilde{x})^T \tilde{\theta} B^T P \tilde{x} + \eta^T B^T P \tilde{x} + \frac{1}{\gamma} tr(\dot{\hat{\theta}}^T \tilde{\theta}) \qquad (19)\end{aligned}$$

Noting that

$$\phi(\tilde{x})^T \tilde{\theta} B^T P\tilde{x} = tr(B^T P\tilde{x}\phi(\tilde{x})^T \tilde{\theta})$$

if we choose the parameter update law as

$$\dot{\hat{\theta}} = \gamma\phi(\tilde{x})\tilde{x}^T PB - c_0\gamma \frac{\tilde{x}PB\hat{\theta}^T\phi(\tilde{x})}{M_\theta^2}\hat{\theta}\,, \qquad (20)$$

$$c_0 = \begin{cases} 1 & if\ \|\hat{\theta}^T\| = M_\theta\ and\ \tilde{x}^T PB\hat{\theta}^T\phi > 0 \\ 0 & otherwise \end{cases} \qquad (21)$$

It can be easily verified that if the initial parameters are chosen to be inside the ball, i.e., $\|\hat{\theta}(0)\| := \{tr[\hat{\theta}(0)^T\hat{\theta}(0)]\}^{1/2} \le M_\theta$, then we have $\|\hat{\theta}(t)\| \le M_\theta$ for all $t \ge 0$. Then we have

$$\dot{V} = -\frac{1}{2}\tilde{x}^T Q\tilde{x} + \frac{1}{\gamma}tr(-\gamma B^T P\tilde{x}\phi(\tilde{x})^T\tilde{\theta} + \dot{\hat{\theta}}^T\tilde{\theta}) + \eta^T B^T P\tilde{x}$$

$$= -\frac{1}{2}\tilde{x}^T Q\tilde{x} + \frac{1}{\gamma}tr[-\gamma\phi(\tilde{x})\tilde{x}^T PB + \dot{\hat{\theta}})^T\tilde{\theta}] + \eta^T B^T P\tilde{x}$$

$$= -\frac{1}{2}\tilde{x}^T Q\tilde{x} + \eta^T B^T P\tilde{x} - c_0 tr(\frac{\tilde{x}^T PB\hat{\theta}^T\phi}{M_\theta^2}\tilde{\theta}^T\hat{\theta}) \qquad (22)$$

Now let's have a look at the last term in the above equation,

$$c_0 tr(\frac{\tilde{x}^T PB\hat{\theta}^T\phi}{M_\theta^2}\tilde{\theta}^T\hat{\theta})$$

$$= c_0\frac{\tilde{x}^T PB\hat{\theta}^T\phi}{M_\theta^2}tr(\tilde{\theta}^T\hat{\theta})$$

$$= c_0\frac{\tilde{x}^T PB\hat{\theta}^T\phi}{M_\theta^2}tr(\hat{\theta}^T\hat{\theta} - \theta^{*T}\hat{\theta})$$

$$\ge 0$$

Actually, it is noted that when $\|\hat{\theta}\| < M_\theta$ or $\|\hat{\theta}\| = M_\theta$ and $\tilde{x}^T PB\hat{\theta}^T\phi \le 0$, then $c_0 = 0$, the above inequality is trivial; when $\|\hat{\theta}\| = M_\theta$ and $\tilde{x}^T PB\hat{\theta}^T\phi > 0$, then, due to the fact that $\|\theta^*\| \le M_\theta$, we can have $(\hat{\theta}^T\hat{\theta} - \theta^{*T}\hat{\theta}) \ge 0$, therefore, the above inequality is also true. In other words, the projection will not make the derivative of the Lyapunov type function more positive. Therefore we have

$$\dot{V} \le -\frac{1}{2}\tilde{x}^T Q\tilde{x} + \eta^T B^T P\tilde{x}$$

$$\le -\frac{1}{2}\lambda_{\min}(Q)\|\tilde{x}\|^2 + \eta_0\lambda_{\max}(P)\|B\|\|\tilde{x}\| \qquad (23)$$

$$= -\frac{1}{2}\|\tilde{x}\|[\lambda_{\min}(Q)\|\tilde{x}\| - 2\sqrt{2}\,\eta_0\lambda_{\max}(P)]$$

where $\lambda_{max}(P)$ and $\lambda_{mix}(Q)$ denote the maximum eigenvalue of matrix P and the minimum eigenvalue of matrix Q respectively.

Consequently, it can be concluded from the eqn.(23) that the system is convergent and tracking error $\tilde{x}$ will belong to a residue of radius $R_0 = \beta\eta_0$ with $\beta := \frac{2\sqrt{2}\ \lambda_{\max}(P)}{\lambda_{\min}(Q)}$.

This implies that the positioning error will also belong to the residue.

4. Conclusions

A new stepper motor control scheme is developed in this paper. The proposed scheme consists of a conventional feedback linearization controller, which is based on the known nominal stepper motor dynamics model, and a compensating controller, which is based on the RBF neural network. The compensating controller is used to improve the stepper motor positioning performance. The neural network is trained on-line based on Lyapunov theory and learning convergence is thus guaranteed.

References

[1] M. Zribi and J. Chiasson, "Position control of a PM stepper motor by exact linearization", *IEEE Trans. Automat. Contr.*, vol.AC-36, pp.620-625, 1991.

[2] M. Ilic-Spong, R. Marino, S. Peresada, and D.G. Taylor, “Feedback linearizing control of switched reluctance motors”, *IEEE Trans. Automat. Contr.*, vol.AC-32, no.5, 1987.

[3] D.G. Taylor, M. Ilic-Spong, and S. Peresada, “Nonlinear composite control of switched reluctance motors”, *Proc. 12th IEEE Conf. Industrial Electronics*, Milwaukee, WI, Oct. 1986.

[4] J.B.D. Cabrera and K.S. Narendra, “Issues in the application of neural networks for tracking based on inverse control”, *IEEE Trans. Automat. Contr.*, vol.44, pp.2007-2027, 1999.

[5] S. Jagannathan and F.L. Lewis, “Discrete-time neural net controller for a class of nonlinear dynamical systems”, *IEEE Trans. Automat. Contr.*, vol.41, pp.1693-1699, 1996.

[6] R.M. Sanner and J.J.E. Slotine, “Gaussian networks for direct adaptive control”, IEEE Trans. Fuzzy Systems, vol.3, pp.837-863, 1992.

[7] T. Kenjo, *Stepping motors and their microprocessor controls*. Oxford, U.K., Clarendon, 1984.

[8] E.J. Hartman, J.D. Keeler and J.M. Kowalski, "Layered neural networks with gaussian hidden units as universal approximations", *Neural Computation*, vol.2, pp. 210-215, 1990.

[9] J. Park and I.W. Sandberg, "Universal approximation using radial-basis-function networks", *Neural Computation*, vol.3, pp.246-257, 1990.

[10] R.M. Sanner and J.J.E. Slotine, "Gaussian networks for direct adaptive control", *Proc. American Control Conf.* pp.2153-2159, 1991.

[11] T. Holcomb and M. Morari, "Local training of radial basis function networks: towards solving the hidden unit problem", *Proc. American Control Conf.*, pp.2331-2336, 1991.