Nonlinear modeling and control approach to magnetic levitation ball system using functional weight RBF network-based state-dependent ARX model

Yemei Qin\textsuperscript{a,b}, Hui Peng\textsuperscript{a,b,\ast}, Feng Zhou\textsuperscript{a,b}, Xiaoyong Zeng\textsuperscript{a,b}, Jun Wu\textsuperscript{b,c}

\textsuperscript{a}School of Information Science & Engineering, Central South University, Changsha, Hunan 410083, China
\textsuperscript{b}Collaborative Innovation Center of Resource-conserving & Environment-friendly Society and Ecological Civilization, Changsha, Hunan 410083, China
\textsuperscript{c}School of Electrical & Information Engineering, Changsha University of Science & Technology, Changsha, Hunan 410004, China

Received 24 December 2014; received in revised form 24 April 2015; accepted 3 June 2015
Available online 2 July 2015

Abstract

A hybrid model, which adopts a radial basis function (RBF) neural networks with functional weights (FWRBF) to approximate the coefficients of the state-dependent AutoRegressive model with eXogenous input variables (SD-ARX), is built for modeling a magnetic levitation ball system and is referred to as the functional weight RBF nets-based ARX (FWRBF-ARX) model. This model structure, which may be identified by using the historical input/output data, inherits both the advantages of the FWRBF networks in function approximation and of the state-dependent ARX models in description of nonlinear dynamics. Due to the structured characteristics of the FWRBF-ARX model, an offline structured nonlinear parameter optimization method (SNPOM) is applied to identify the model structure and parameters. Using the input and output observation data of the real system, a FWRBF-ARX model with small residual, small standard variation and small long-term predictive residual can be identified. Based on the local linearity of the built FWRBF-ARX model at certain working point, a locally linearized model predictive controller (MPC) is designed to achieve stable levitating and output-tracking control of the steel ball in the electromagnetic field. From the real-time control results, it is seen that the FWRBF-ARX model-based MPC may control the steel
ball to track step signals very well, and may obtain much better control performance within wide step range compared to conventional PID control, the ARX model and RBF-ARX model-based MPCs.

1. Introduction

Magnetic levitation (maglev) ball systems are inherently nonlinear and open-loop instable, so that they are often used as a benchmark experimental platform for testing advanced control strategies [1,2]. The application of maglev technology requires that a stable output-tracking control for the object is necessary. Recently, a variety of control strategies have been presented and discussed for the levitation system. For instance, the feedback linearization [3] or feedforward linearization [4] technology was applied to design a trajectory tracking controller for a nonlinear maglev system. Fuzzy-PID control [5] and adaptive control [3,6,7], sliding-model control [8], fuzzy control [9,10], neural network control [11] and predictive control [1,12–14] were also used to achieve the trajectory tracking in order to improve the robustness and to expand the range of effective control. However, these approaches are mostly on the basis of the physical model of a maglev system. Due to the effects of magnetic saturation and eddy current, it may be very difficult to acquire an accurate physical model [15] to catch the dynamic behavior of a maglev system.

Therefore, some approaches are considered to approximate its nature behavior. The most common way is to give some assumptions and then to acquire a simplified electrical and mechanical model of the underlying system under these conditions [1,6,16]. However, some parameters in the simplified model may be still unknown and not easy to be confirmed through its internal structure. Baechle et al. [1] identified unknown parameters of the electrical and mechanical subsystems of a maglev system by using a nonlinear least-squares solver. Morales et al. [6] estimated unknown parameters online by adopting a nonlinear algebraic identification technique. Glueck et al. [16] analyzed the properties of a maglev system and derived a detailed physical model, then estimated the inductance and the position of a sensor-less maglev system. These methods mentioned above generally require that the structure of the model is known and that only several parameters need to be estimated, which are a physically motivated gray-box approach.

If the real input and output data of a controlled object may be obtained, in fact, it is an advisable choice to apply a nonlinear system identification approach, i.e. a historical data-driving black-box approach. One advantage of the data-driving modeling approach to a system is that it is unnecessary to precisely analyze the internal structure and the relationship between physical variables. Generally, it necessitates an accurate model that can capture the nonlinearities of the unstable maglev system if a high control performance and a wide operational region are desired. Neural networks [11,17] may offer a framework for the black-box modeling of the complex nonlinear maglev system, because of their simple topological structure and their precision in nonlinear dynamics approximation. However, in many real applications, a large number of network parameters may lead to difficulties in parameter estimation. A state-dependent parameter (SDP) model with linear wavelet function was built in [18] and excellently characterized the magnetic bearing system’s nonlinear dynamics. A state-dependent ARX model using a set of radial basis function (RBF) networks to approximate the coefficients of nonlinear ARX model was also proposed in [19] and captured the dynamic feature of a maglev ball system well.
In order to more accurately capture and quantify the maglev system's dynamics on the basis of the previous work [19], this study further explores to build an enhanced version of the state-dependent ARX model, which uses RBF nets with state-dependent functional weights (FWRBF) to approximate the state-dependent coefficients of the nonlinear ARX model. In the FWRBF networks, the connection weights between the output layer and the hidden layer are linear functions depending on system's working point state, instead of being constant weights in a normal RBF network. Rojas et al. [20] replaced the constant weights of RBF networks with regression weights functions of the input variables, and showed good approximation performance of the improved model; Gan et al. [21] applied the linear regressive weights into RBF-AR model to predict a chaotic time series and achieved high predictive precision. This direct link between the weights and the input variables makes the RBF network more efficient and more flexible, and may need few nodes in the network. In this paper, a function of the working-point state is used to approximate the weights of RBF networks in the RBF-ARX model, and then a hybrid model which is referred to as the functional weight RBF nets-based ARX (FWRBF-ARX) model is built. This model belongs to a class of globally nonlinear models due to the time-varying working-point of system, and is also a set of locally linear models for fixed working-points. This modeling method inherits the advantages of the state-dependent ARX models in description of nonlinear dynamics and of the FWRBF networks in function approximation. This model structure may improve the model's ability of describing the maglev system locally as well as globally and significantly reduce the number of hidden nodes in RBF networks.

This paper presents the origin of the FWRBF-ARX model for a nonlinear system, and builds a FWRBF-ARX model for a maglev system in Section 2. In Section 3, based on the built FWRBF-ARX model, a locally linear predictive controller is designed to make the ball levitate stably in the magnetic field. Section 4 gives the real-time control results, and compares four control methods (the proposed FWRBF-ARX model-based MPC, RBF-ARX model-based MPC [19], linear ARX model-based MPC and classical PID control) for the underlying system in order to verify the effectiveness and superiority of the proposed modeling and control approach. The computation complexity is discussed in Section 5. Finally in Section 6, the conclusions are outlined.

2. Functional weight RBF network-based state-dependent ARX (FWRBF-ARX) model

At the seminar on the discipline development strategy of Control Science of China in 2013, Academician Huang Lin of Chinese Academy of Sciences pointed out that ‘Research on the model of a system is the primary task of control science’ [22]. It is clear that building a model, which can exactly represent the dynamics of a maglev system to be studied in this paper, is a very important work of improving control performance.

Real time sampled input and output data of a system may contain all internal and external information, therefore a model identification approach using these observation data is often an optional method, especially for those systems whose accurate physical models are difficult to be acquired, and whose working-points are time-variant. When a system is excited persistently within a wide range and its dynamic modes are motivated sufficiently, the input/output data-based modeling may capture the dynamic characteristics of the system. This work will apply the system identification technology to model an experimental maglev system.

2.1. FWRBF-ARX model

In order to represent dynamic behavior of a nonlinear system, we need to build a nonlinear model. An important class of discrete-time nonlinear model is the nonlinear AutoRegressive...
model with eXogenous variables (NARX) model \[23\] as follows

\[ y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u)) + \xi(t) \] (1)

where \( f(\bullet) \) denotes a nonlinear structure, \( n_y \) and \( n_u \) are the order of output variable \( y \) and input variable \( u \), respectively. Let \( W(t-1) = [y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u)]^T \) and assume that the nonlinear function \( f(\bullet) \) in (1) is continuously differentiable at a working-point \( W_0 \), then one may expand \( f(\bullet) \) in a Taylor series about \( W_0 \)[24], yielding

\[ y(t) = \phi_0(W(t-1)) + \sum_{i=1}^{n_y} \phi_{y,i}(W(t-1))y(t-i) + \sum_{i=1}^{n_u} \phi_{u,i}(W(t-1))u(t-i) + \xi(t) \] (2)

where \( \phi_0 \), \( \phi_{y,i} (i = 1, 2, \ldots, n_y) \) and \( \phi_{u,i} (i = 1, 2, \ldots, n_u) \) are regressive coefficients after Taylor expansion. Model (2) has a structure similar to linear ARX model and is also a kind of historical input/output data-driving model. The coefficients of model (2) can vary with the working-point of the system. If these coefficients become constant, the model is just a linear ARX model and may be used as a linearization model of model (2), which may catch the nonlinear systems' local dynamics within a small range. One may use some functions, such as polynomial \[25\], fuzzy-neural networks \[26\] or RBF neural networks \[19,27\], to approximate the variable coefficients of each regressive term in model (2). When these approximation functions depend on the working-point state, the model (2) is then called as the state-dependent ARX model.

RBF neural network is an efficient tool of solving nonlinearity approximation problem because of its strong approximation ability and simple topological structure. The RBF network weights linking hidden layer unit and output layer unit are usually constant, so they require many nodes in hidden layer unit to attain high approximation precision. In order to reduce the number of the nodes, Rojas et al. \[20\] replaced the constant weights of RBF networks with regression weights which are functions of the input variables and obtained good approximation performance. Gan et al. \[21\] applied the linear regressive weights into a RBF-AR model to predict a chaotic time series and achieved high predictive precision. This method was also applied into the wavelet neural networks (WNN) \[28\] and reduced the number of hidden neurons.

In order to obtain better approximation of the nonlinear functions, i.e. the functional coefficients in model (2), an improved RBF network called a state-dependent functional weight-RBF network (FWRBF) is proposed to approximate the coefficients \( \phi_0 \), \( \phi_{y,i} \) and \( \phi_{u,i} \) of model

![Fig. 1. RBF network with state-dependent linear function weights.](image-url)
The FWRBF network uses a linear function depending on the working-point state to replace constant weights of a conventional RBF networks. For the sake of simplicity and without loss of generality, we assume that there is only one output signal; the structure of the improved RBF network is shown in Fig.1.

The output \( \phi \) in Fig.1 is

\[
\phi = c_0 + \sum_{k=1}^{m} (v_{k,0} + V_k^T W) \varphi(W, Z_k)
\]

\[
= c_0 + \sum_{k=1}^{m} (v_{k,0} + v_{k,1}w_1 + \ldots + v_{k,n_w}w_{n_w}) \varphi(W, Z_k)
\]

(3)

where, \( \varphi(W, Z_k) \) is a radial basis function, \( W = (w_1, w_2, \ldots, w_{n_w})^T \) is the working-point state, \( m \) is the number of nodes of hidden layer, \( n_w \) is the dimension of \( W, Z_k \) \((k = 1, 2, \ldots, m)\) are the center of RBF networks, \( v_k = v_{k,0} + v_{k,1}w_1 + \ldots + v_{k,n_w}w_{n_w} \) \((k = 1, 2, \ldots, m)\) is the output weight which varies with \( W \) and makes the RBF networks more flexible, \( v_{k,i} \) \((i = 0, 1, \ldots, n_w)\) is the scaling coefficient, \( c_0 \) is the bias.

Next, the FWRBF networks (3) is introduced into model (2) to approximate the functional regressive coefficients \( \phi_0, \phi_{y,i} \) and \( \phi_{u,i} \). Thus a FWRBF-ARX model, which has functional weight coefficients depending linearly on the working-point state, is constructed. The structure of single-input-single-output (SISO) FWRBF-ARX model is presented in Eq. (4) where the Gaussian function is selected as the radial basis function, and \( \lambda \) is a scaling factor.

\[
y(t) = \phi_0(W(t-1)) + \sum_{i=1}^{n_y} \phi_{y,i}(W(t-1))y(t-i) + \sum_{i=1}^{n_u} \phi_{u,i}(W(t-1))u(t-i) + \xi(t)
\]

\[
\begin{align*}
\phi_0(W(t-1)) &= c_0^0 + \sum_{k=1}^{m} (v_{k,0}^0 + v_{k,1}^0w_1 + \ldots + v_{k,n_w}^0w_{n_w}) \exp\left(-\lambda_k^2 \|W(t-1) - Z_k^0\|^2\right) \\
\phi_{y,i}(W(t-1)) &= c_{i,0}^y + \sum_{k=1}^{m} (v_{k,0}^y + v_{k,1}^yw_1 + \ldots + v_{k,n_w}^yw_{n_w}) \exp\left(-\lambda_k^2 \|W(t-1) - Z_k^y\|^2\right) \\
\phi_{u,i}(W(t-1)) &= c_{i,0}^u + \sum_{k=1}^{m} (v_{k,0}^u + v_{k,1}^uw_1 + \ldots + v_{k,n_w}^uw_{n_w}) \exp\left(-\lambda_k^2 \|W(t-1) - Z_k^u\|^2\right) \\
W(t-1) &= [w_1 \ w_2 \ \ldots \ \ w_{n_w}]^T
\end{align*}
\]

(4)

In model (4), \( y(t) \) is the output variable, \( u(t) \) is the input variable, \( \|\cdot\| \) denotes 2-norm, \( \phi_0, \phi_{y,i}(i = 1, 2, \ldots, n_y) \) and \( \phi_{u,i}(i = 1, 2, \ldots, n_u) \) are the Gaussian nonlinear state-dependent coefficients and vary with the working-point state, \( W(t-1) \) is time-varying and may be input, output, any other measureable signal or their combination. The FWRBF-ARX model possesses an autoregressive structure which is similar to a linear ARX structure and partially disperses the complexity of the model into the AutoRegressive part, so the selection of centers of RBF networks may be relaxed. Although all RBF networks in (4) share the same center, the regressive functional coefficients may be different. The regressive weights \( v_k \) depend on the working-point \( W(t-1) \) of system at time \( t-1 \) and are multiplied by the locally Gaussian radial basis function, thus, model (4) may reflect more local information of a system. If \( W(t-1) \) is fixed, model (4) is only a simple linear ARX model and may reflect locally linear behavior of a system, which may be regarded as a local linearization of the system at a working-point. This property is very useful
and may convert a nonlinear model-based control into a linear model-based control. On the other hand, model (4) also represents the system’s globally nonlinear nature when the working-point changes with the evolution of the process. Moreover, the improved RBF network structure may greatly reduce the number of networks and/or centers of RBF networks [21], as well as the computation time.

2.2. Identification of FWRBF-ARX model

To identify the FWRBF-ARX model, we need to estimate the structure and parameters of the model. The structure of model (4) is determined by model orders and nonlinear parameters to be estimated in model (4). The number of linear parameters is much larger than that of nonlinear parameters, so it is very suitable to apply SNPOM to identify parameters of the model (4). The optimization process is given as follows.

Step 1. Parameter classification

For the FWRBF-ARX model (4), the parameters to be estimated are divided into two subspaces: the linear space \( \theta_L = \{ c^0_0, c^0_{i,j}, v^0_{k,0}, v^j_{k,l} | k = 1, ..., m; \ i = 1, ..., n_y; \ j = y \ or \ u; \ l = 0, ..., n_w \} \) and the nonlinear space \( \theta_N = \{ \lambda^0_k, \lambda^j_k | k = 1, ..., m; \ j = y \ or \ u \} \). This type of model may be identified by applying the structured nonlinear parameter optimization method (SNPOM) which is proposed for neural network-type models [29] to identify all the parameters. The SNPOM estimates linear and nonlinear parameters separately and alternately, and uses the Levenberg-Marquardt method (LMM) for nonlinear parameters and the least-squares method (LSM) for linear parameters. Compared with general non-structured gradient search methods, the SNPOM is more flexible, and converges faster and the computation is also more efficient, especially for systems that have more linear parameters than nonlinear ones. Moreover, it is an offline optimization method and is suitable for identifying the model driven by historical data. There are \((1 + m (n_w + 1)) (n_y + n_u + 1)\) linear parameters and \((m + m \times n_w)\) nonlinear parameters to be estimated in model (4). The number of linear parameters is much larger than that of nonlinear parameters, so it is very suitable to apply SNPOM to identify parameters of the model (4). The optimization process is given as follows.

\[
y(t) = f(\theta_L, \theta_N, W(t-1)) + \xi(t) \quad (5)
\]

or

\[
y(t) = \Psi(\theta_N, W(t-1))^T \theta_L + \xi(t) \quad (6)
\]

Model (6) is the regression form of model (5), and is also linear with respect to \( \theta_L \).

Step 2. Initialization

First, the orders of model need to be chosen. For an FWRBF-ARX model (4), the orders include \( n_y, n_u, m \) and \( n_w \). An appropriate order is determined in Step 4.

Second, the initial values \( \theta_L^0 \) are chosen from prior knowledge of the system. In this paper, we randomly choose a subset \( Z^0_k (k = 1, ..., n_w; j = y \ or \ u) \) in the vector values \( W(t-1) \) as the initial values for the FWRBF network centers and then use the following formula to compute the initial scaling parameters \( \lambda^0_k (k = 1, ..., n_w; j = y \ or \ u) \).

\[
\lambda^0_k = - \log \varepsilon_k / \max_{i=1}^{\| W(t-1) - Z^0_k \|^2_2}
\]

4338

where, $\varepsilon_k \in [0.0001, 0.1]$. Thus, the linear parameters will be bounded and stable when the signal $W(t-1)$ moves far away from the centers.

Next, fix $\theta^0_N$ and estimate the initial linear parameters $\theta^0_L$ by LSM:

$$
\theta^0_L = \left[ R(\theta^0_N)^T R(\theta^0_N) \right]^{-1} R(\theta^0_N)^T \bar{Y}
$$

where

$$
R(\theta^0_N) = \begin{pmatrix}
\Psi(\theta^0_N, W(\tau))^T \\
\Psi(\theta^0_N, W(\tau + 1))^T \\
\vdots \\
\Psi(\theta^0_N, W(M - 1))^T
\end{pmatrix}
$$

$$
\bar{Y} = (\bar{y}(\tau + 1) \quad \bar{y}(\tau + 2) \quad \ldots \quad \bar{y}(M))
$$

Step 3. Optimization

The objective function is taken to be the sum of squares of the model residuals and is given as follows:

$$
V(\theta_L, \theta_N) = \frac{1}{2} || F(\theta_L, \theta_N) ||_2^2
$$

$$
F(\theta_L, \theta_N) = \begin{pmatrix}
\hat{y}(\tau + 1 | \tau) - \bar{y}(\tau + 1) \\
\hat{y}(\tau + 2 | \tau + 1) - \bar{y}(\tau + 2) \\
\vdots \\
\hat{y}(M | M - 1) - \bar{y}(M)
\end{pmatrix}
$$

where $\hat{y}(\tau + 1 | \tau)$ is the one-step-ahead prediction of the output based on model (5). The optimization problem is to compute

$$
(\hat{\theta}_L, \hat{\theta}_N) = \arg\min_{\theta_L, \theta_N} V(\theta_L, \theta_N)
$$

The optimization iteration is divided into the following two steps:

First, the updating formula of the nonlinear parameters is

$$
\theta^{k+1}_N = \theta^k_N + \beta_k d_k
$$

where $\kappa(k = 0, 1, 2, \ldots, \kappa_{\text{max}})$ denotes the iteration step, $d_k$ is the search direction, and $\beta_k$ is a scalar step length parameter which represents the distance to the minimum. In order to increase the robustness of the search process using the LMM, $d_k$ in Eq. (11) is obtained from a solution to the set of linear equations as follows:

$$
\left[ J(\theta^k_N)^T J(\theta^k_N) + \gamma_k I \right] d_k = - J(\theta^k_N)^T F(\theta^k_L, \theta^k_N)
$$

where $J(\theta^k_N) = \left( \partial F(\theta^k_L, \theta^k_N) / \partial \theta^k_N \right)^T$ is Jacobian matrix of $F(\theta^k_L, \theta^k_N)$ with respect to $\theta^k_N$, the scalar $\gamma_k$ determines both the magnitude and direction of $d_k$. When $\gamma_k$ tends to infinity, $d_k$ will tend towards the steepest descent direction. As $\gamma_k$ tends to zero, $d_k$ tends towards the Gauss–Newton direction. The size of $\gamma_k$ is altered at each iteration by using a method similar to that used in the function ‘lsqnonlin’ in the Matlab Optimization Toolbox. $\beta_k$ in Eq. (11) is then calculated by a
line search procedure, such as the mixed quadratic and cubic polynomial interpolation and extrapolation method.

Secondly, after calculating $\theta_{N}^{\kappa+1}$ using Eq. (11) at each iteration, the linear weights $\theta_{L}^{\kappa+1}$ are immediately updated using the LSM as follows.

$$\theta_{L}^{\kappa+1} = \left[ R(\theta_{N}^{\kappa+1})^T R(\theta_{N}^{\kappa+1}) \right]^{-1} R(\theta_{N}^{\kappa+1})^T \bar{Y}$$

$$R(\theta_{N}^{\kappa+1}) = \begin{pmatrix}
\Psi(\theta_{N}^{\kappa+1}, \bar{W}(\tau))^T \\
\Psi(\theta_{N}^{\kappa+1}, \bar{W}(\tau + 1))^T \\
\vdots \\
\Psi(\theta_{N}^{\kappa+1}, \bar{W}(M-1))^T
\end{pmatrix}$$

$$\bar{Y} = (\bar{y}(\tau + 1), \bar{y}(\tau + 2), \ldots, \bar{y}(M))$$

(13)

The termination condition is

$$V(\theta_{L}^{\kappa+1}, \theta_{N}^{\kappa+1}) < V(\theta_{L}^{\kappa}, \theta_{N}^{\kappa})$$

(14)
at each iteration. Hence, $\theta_{N}^{\kappa+1}$ and $\theta_{L}^{\kappa+1}$ are the best parameter choices to decrease the objective function (9) at the $(\kappa + 1)$th iteration.

**Step 4. Selecting the order of the model**

To the estimation of model orders, we mainly use the Akaike information criterion (AIC) as evaluation standard. For the FWRBF-ARX model (4), the AIC is defined as follows:

$$\text{AIC} = M \log \hat{\sigma}_e^2 + 2(s + 1)$$

(15)

where $s$ denotes the total number of parameters to be estimated. In the FWRBF-ARX model (4), $s = (1 + m \ (n_w + 1)) (n_y + n_u + 1) + (m + m \times n_w)$; $\hat{\sigma}_e^2$ is the model's one-step-ahead prediction variance under the chosen orders $(n_y, n_u, m, n_w)$. Note that, the number of observation data must be much greater than the largest order of the model, i.e. $M \geq \max(n_y, n_u, m, n_w)$. The second part of Eq. (15) is a penalty term to high order model with same residuals.

The procedure is to repeat the above Step 3 for different orders and then choose the best model by looking for a small AIC value. Simultaneously, we also synthetically consider the long-term prediction residuals and the dynamic characteristics of step response of the estimated model. By repetitively training the models with different orders, finally, the selected best model should have small AIC value, small modeling residual and good dynamic performance.

**2.3. Modeling and identification of the maglev system**

An experimental maglev system in Fig. 2 is used as the research object in this work. This maglev device which is linked by a control computer consists of an electromagnet, a power amplifier, a photoelectric sensor, a LED light source and a controlled steel ball, as shown in Fig. 2 (a). The experimental device is a single axis of maglev system and only able to control the object to move up and down. This maglev device is a single-input-single-output (SISO) system, and the voltage of driving circuits in the electromagnet is the input signal and the position of the ball is the output signal. Coordinates of the position are depicted in Fig. 2(b) where the electromagnet surface is defined as the origin and the gap between the ball and the electromagnet surface is defined as the position $y$ of the ball, and the ball's controllable interval is from -16 mm to 0 mm in this experimental device. The task of the controller is to make the ball levitate stably...
on a setting position or track a desired trajectory. When the current flows through the coil, whose value is decided by the controller, the electromagnetic attraction $F$ is generated to control the position of the maglev ball.

Due to magnetic saturation effect, the magnetic induction intensity is not proportional to the current of coil when the electromagnet goes into the working state of the magnetic saturation. In addition, due to eddy current effect, the electromagnetic coil may be affected in turn when the ball lies in the magnetic field. They enhances the nonlinearity of the maglev system in Fig. 2 and also make it very difficult to acquire an accurate physical model [15] to catch the dynamics of the maglev system. In order to build the physical model of the system, one has to make several assumptions as follows.

(i) An ideal magnetic field exists between the electromagnet and the ball;
(ii) The ball is a homogeneous sphere;
(iii) Magnetic resistance between the electromagnet and the ball, and the leakage of magnetic flux between windings are both ignored;
(iv) Magnetic saturation and eddy current effect are ignored.

On the basis of Newton's second law, Biot–Savart law, the fundamental law of energy conservation and Kirchhoff's law, the motion equations of the maglev ball system can be given as follows:

$$\begin{align*}
m \frac{d^2 y(t)}{dt^2} &= F(i, y) + mg \\
F(i, y) &= K \left( \frac{di(t)}{dt} \right)^2 \\
u(t) &= Ri(t) + L_1 \frac{di(t)}{dt}
\end{align*}$$

(16)

where $m$ denotes the mass of the steel ball, $y(t)$ is an instantaneous gap between the ball and the electromagnet surface and also denotes the position of the ball, $g$ is the acceleration due to gravity, $F(i, y)$ is the electromagnetic attraction, $u(t)$ is the voltage of electromagnetic coil, $i(t)$ is the instantaneous current through the coil, $R$ is the equivalent resistance of electromagnetic coil, $L_1$ is the self-inductance of the coil, $K$ is a constant coefficient of the electromagnetic coil related to the mutual inductance. The prototype parameters are: $m = 22$ g, $r = 12.5$ mm, $R = 13.8 \ \Omega$, $K = 2.3142 \times 10^{-4}$ Nm$^2$/A$^2$. The detailed derivation of the model (16) can be seen in

![Fig. 2. The maglev ball system. (a) The experimental platform. (b) The principle diagram.](image-url)
If the ball is required to levitate at a point $y_0$, the controller must adjust the control voltage to change the electromagnetic force $F$ at the point $y_0$ and to make the force equal to the gravity $mg$ of the ball, i.e. $F(y_0, t) = mg$. From the physical model (16) one can see that the system has a strong nonlinearity between $y(t)$ and $u(t)$, and is continuously differentiable and may be represented by the FWRBF-ARX model (4).

Based on the proposed modeling method in Section 2.1, the FWRBF-ARX model of the experimental maglev system in Fig. 2 can be built. The maglev system is a SISO system, so its FWRBF-ARX model structure is the same as model (4). In the model, the control voltage of the electromagnetic coil is the input variable $u(t)$, the position of the ball is the output signal $y(t)$, and the output series are set as the working-point state, i.e. $W(t-1) = [y(t-1), \cdots, y(t-n_w)]^T$.

The first step to obtain the FWRBF-ARX model of the maglev system is to generate real-time observation data of the system. The historical data used to identify the FWRBF-ARX model of the maglev system include the position of the steel ball and the voltage of the electromagnetic coil, and they are sampled and converted into digit signals through the PCI1711 converter card which connects the device in Fig. 2 to a Windows XP-based PC with a MATLAB /SIMULINK 2010b environment. Because of the maglev system’s inherent open-loop instability, it is necessary to construct a closed-loop control system in order to attain good levitation effects. In this work, a traditional PID control is applied into the maglev system to obtain the identification data and its block graph is showed in Fig. 3. The ‘sources_data’ module is a given reference signal $y_r$. The ‘PID controller’ is a conventional discrete-time PID controller module provided by MATLAB/SIMULINK, and its control variable is $u(t) = K_p e(t) + K_i \sum_{j=0}^{t} e(j) T + K_d e(t-1) T$, where $e(t) = y_r(t) - y(t)$ denotes the error between the required output $y_r(t)$ and the actual output $y(t)$, $K_p$, $K_i$ and $K_d$ are the controller parameters, and $T$ is the sampling period. The ‘Real Control’ module is the Real-time Windows Target of the maglev object and its internal structure is shown in Fig. 4. The ‘Gain’ is used to convert the output voltage value (unit: V) into the position of ball (unit: mm), which is obtained by calibrating the photoelectric sensor in Fig. 2. The sampling period is set as 5 ms for all experiments. To make the maglev system identifiable and to ensure that its dynamics can be described in a wide range, it is also necessary to make the ball move stably within a range as large as possible. Simultaneously, a white noise signal is added into the input control signal in order to excite the systems’ dynamic modes. In this study, a comprehensive reference trajectory is used and the PID controller parameters are well tuned to make the closed-loop system stable for tracking the required output. The finally determined best PID controller parameters are $K_p = 1.4$, $K_i = 0.04$, $K_d = 18$. A set of observation data of the maglev system under the control of PID is attained, which are shown in Fig. 5. From the data, it is clear that the ball tracked various trajectories, which benefits to stimulate system’s various dynamic modes and to identify its FWRBF-ARX model. In 8000 observed data, the first 4000 data points are used to estimate the parameters of model (4) and the last 4000 data points are used to test the modeling performance.
The second step to obtain the FWRBF-ARX model is to apply the SNPOM and the mentioned model evaluation standard in Section 2.2 to determinate the orders of the model and the parameters. After repetitively training and comparison, the orders of the FWRBF-ARX model of the maglev system are chosen as $n_y = 7$, $n_u = 6$, $m = 1$ and $n_w = 1$. The model output and residuals of one-step-ahead prediction for training data and testing data are shown in Figs. 6 and 7.

From Fig. 6, the one-step-ahead predictive outputs of trained FWRBF-ARX model are almost coincident with the actual output curve. The modeling residuals are very small and the majority of them are within the interval $[-0.04, 0.04]$. Moreover, the residuals show a Gaussian distribution. Fig. 7 also shows that there are very small errors between the test output of the model and the actual output. A small bias exists in the histogram of the residuals in Fig. 7, but it
still shows a Gaussian distribution as a whole. Near to 3500th point in Fig. 6, the modeling errors are a little big, which is caused by small and frequent oscillations here. The modeling results verify that the modeling method for the maglev system is feasible and effective.

2.4. Performance of FWRBF-ARX model

In order to detect the significance of the built FWRBF-ARX model structure, we did a comparison of the proposed model with a linear ARX model and a RBF-ARX model [19] about their approximation ability. The three types of models have the same autoregressive structure, but their approximation functions of regressive coefficients are different. The approximation functions of regressive coefficients of the FWRBF-ARX model and the RBF-ARX model are time-varying and depend on the system's working-point state. In fact, the two models consist of a set of pseudo linear ARX model varying with the working-point, which make it able to capture the dynamics of the nonlinear maglev system, so they may be better than a conventional linear ARX model whose regressive coefficients are constant. Table 1 lists one-step-ahead predictive residuals for the FWRBF-ARX model with different orders, and the results obtained from the linear ARX model and RBF-ARX model are also shown in the same table for comparison. Note that the ARX model and RBF-ARX model are identified through the same SNPOM and observation data shown in Fig. 5. It can be seen that the FWRBF-ARX model produces smaller prediction errors than other two models with the same orders in the training and testing phase. This verifies the effectiveness of introducing the FWRBF network into NARX model to describe the nonlinear maglev system.
In addition, the AIC value is usually used as a criterion to evaluate the modeling result. Table 1 reports the AIC values of three identified models with different orders. One can see that under the condition of same orders, the FWRBF-ARX model has the smallest AIC value. It also

![Fig. 7. Outputs, modeling residuals and histogram of residuals of FWRBF-ARX model for testing data (standard deviation: 0.0108).](image-url)

<table>
<thead>
<tr>
<th>Models($n_y, n_u, m, n_w$)</th>
<th>Number of centers</th>
<th>Total of unknown parameters</th>
<th>Training data</th>
<th>Testing data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Predictive error variance</td>
<td>AIC</td>
</tr>
<tr>
<td>ARX(6,2)</td>
<td>–</td>
<td>9</td>
<td>$3.7273 \times 10^{-4}$</td>
<td>$-3.1483 \times 10^{4}$</td>
</tr>
<tr>
<td>RBF-ARX(6,2,1,2)</td>
<td>4</td>
<td>18</td>
<td>$2.6181 \times 10^{-4}$</td>
<td>$-3.2894 \times 10^{4}$</td>
</tr>
<tr>
<td>FWRBF-ARX (6,2,1,2)</td>
<td>4</td>
<td>36</td>
<td>$2.4034 \times 10^{-4}$</td>
<td>$-3.3250 \times 10^{4}$</td>
</tr>
<tr>
<td>ARX(7,6)</td>
<td>–</td>
<td>14</td>
<td>$3.5587 \times 10^{-4}$</td>
<td>$-3.1644 \times 10^{4}$</td>
</tr>
<tr>
<td>RBF-ARX(7,6,1,1)</td>
<td>2</td>
<td>28</td>
<td>$2.5404 \times 10^{-4}$</td>
<td>$-3.2990 \times 10^{4}$</td>
</tr>
<tr>
<td>FWRBF-ARX (7,6,1,1)</td>
<td>2</td>
<td>42</td>
<td>$1.9355 \times 10^{-4}$</td>
<td>$-3.4044 \times 10^{4}$</td>
</tr>
<tr>
<td>ARX(10,7)</td>
<td>–</td>
<td>18</td>
<td>$3.5114 \times 10^{-4}$</td>
<td>$-3.1654 \times 10^{4}$</td>
</tr>
<tr>
<td>RBF-ARX(10,7,1,2)</td>
<td>4</td>
<td>36</td>
<td>$2.2741 \times 10^{-4}$</td>
<td>$-3.3387 \times 10^{4}$</td>
</tr>
<tr>
<td>FWRBF-ARX (10,7,1,2)</td>
<td>4</td>
<td>72</td>
<td>$2.0959 \times 10^{-4}$</td>
<td>$-3.3726 \times 10^{4}$</td>
</tr>
</tbody>
</table>

Table 1

Comparison results of one-step prediction for different models.
provides an evidence for the significance of introducing the state-dependent functional weights to approximate the weights in RBF networks and constructing the FWRBF-ARX model for the maglev system. Moreover, the linear ARX model has the largest AIC value compared with the others; it explains the ARX model’s weak ability to describe dynamics of the nonlinear maglev system. The identified FWRBF-ARX model with orders (7, 6, 1, 1) has the smallest AIC value for the training data and the testing data compared to the others. Therefore, the orders of the FWRBF-ARX model are determined as (7, 6, 1, 1) that is mentioned in Section 2.3. Meanwhile, the linear ARX(10, 7) and the RBF-ARX(10, 7, 1, 2) have the smallest AIC value for ARX or RBF-ARX type model from Table 1, which will be applied into the MPC algorithm presented in Section 4 in order to compare with the FWRBF-ARX model-based MPC.

The long-term prediction error of a model is also often used as an evaluation standard of modeling precision. In order to exhibit the prediction ability of the FWRBF-ARX model, the multi-step-ahead predictive standard deviations (STD) of three models are shown in Table 2, where the model output predictions are calculated from one-step-ahead to 12 steps-ahead predictions. From Table 2, we can see that with the increase of the prediction step, the predictive accuracy of all models declines, but the FWRBF-ARX model shows better modeling precision than the other models. All results given above show the good statistical properties of the estimated FWRBF-ARX model and prove that the estimated FWRBF-ARX model can capture and quantity the dynamic behavior of the maglev ball system quite well.

### 3. FWRBF-ARX model-based predictive control

Because of the online solution of an optimal control problem on a receding horizon and the diversity of selectable model structure, the model predictive control (MPC) method is suitable to implement in practice. MPC has been applied into controlling a nonlinear maglev system [14]. However, the nonlinear optimization control for a nonlinear system is a non-convex programming problem, which will result in high computational burden. This is a drawback of MPC especially for a fast-responding nonlinear system. Kemih et al. [14] applied a constrained generalized predictive control (GPC) based on a linearized model to ensure the stable control of

#### Table 2
Comparison results of multi-step-ahead prediction for different models.

<table>
<thead>
<tr>
<th>Prediction step</th>
<th>Standard deviation (STD)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARX model (10, 7)</td>
</tr>
<tr>
<td>1</td>
<td>0.0083</td>
</tr>
<tr>
<td>2</td>
<td>0.0146</td>
</tr>
<tr>
<td>3</td>
<td>0.0234</td>
</tr>
<tr>
<td>4</td>
<td>0.0348</td>
</tr>
<tr>
<td>5</td>
<td>0.0491</td>
</tr>
<tr>
<td>6</td>
<td>0.0664</td>
</tr>
<tr>
<td>7</td>
<td>0.0869</td>
</tr>
<tr>
<td>8</td>
<td>0.1110</td>
</tr>
<tr>
<td>9</td>
<td>0.1391</td>
</tr>
<tr>
<td>10</td>
<td>0.1720</td>
</tr>
<tr>
<td>11</td>
<td>0.2104</td>
</tr>
<tr>
<td>12</td>
<td>0.2550</td>
</tr>
</tbody>
</table>
the maglev system and to reduce computation burden online. Ulbig et al. [12] built a piecewise affine (PWA) model of a maglev system and constructed a explicit nonlinear predictive control law in order to reduce the computation time and to improve the performance of real-time control. Baechle et al. [1] utilized a tailored gradient optimization method to enhance stability and computation efficiency. A time-varying locally linearized model-based predictive controller, which depends on the variable working-point, was designed to achieve a wide range step response of the maglev objects in [19]. In this work, based on the proposed FWRBF-ARX model, a predictive controller is designed and implemented to an experimental maglev system.

This section describes a nonlinear MPC scheme for a SISO experimental magnetic levitation system with the maximum levitation range of 16 mm. Due to the fast-response requirement of the maglev system and the locally linear, globally nonlinear characteristics of the FWRBF-ARX model, a locally linear prediction controller is designed for the maglev system.

To convert model (4) into a SISO state space model, a state vector is defined as follows.

\[
x(t) = \begin{bmatrix} x_{1,t} & x_{2,t} & \cdots & x_{k_n,t} \end{bmatrix}^T
\]

\[
x_{1,t} = y(t)
\]

\[
x_{k,t} = \sum_{i=1}^{k_n-k+1} \phi_{y,i+k-1}(W(t-1))y(t-i) + \sum_{i=1}^{k_n-k+1} \phi_{u,i+k-1}(W(t-1))u(t-i)
\]

\[
k = 2, 3, \ldots, k_n; \ k_n = \max(n_y, n_u)
\]

Then a state space model from model (4) may be represented:

\[
\begin{cases}
x(t+1) = A_t x(t) + B_t u(t) + \Phi_t + \Xi(t+1) \\
y(t) = C x(t)
\end{cases}
\]

where

\[
A_t = \begin{bmatrix} \phi_{y,1} & 1 & 0 & 0 & 0 \\ \phi_{y,2} & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{y,k_n-1} & 0 & 0 & 0 & 1 \\ \phi_{y,k_n} & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_t = \begin{bmatrix} \phi_{u,1} \\ \phi_{u,2} \\ \vdots \\ \phi_{u,k_n-1} \\ \phi_{u,k_n} \end{bmatrix}
\]

\[
\Phi_t = \begin{bmatrix} \phi_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \phi_0 \end{bmatrix}, \quad \Xi(t+1) = \begin{bmatrix} 1 \cdot (t+1) \\ \vdots \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1^T \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\]

Model (18–19) is a state-space representation of FWRBF-ARX model (4). Note that, \((\bullet) = (W(t))\) denotes that the coefficients is relevant to the working-point state \(W(t)\). Therefore, for a fixed working-point, a locally linear ARX model is obtained from the globally nonlinear FWRBF-ARX model, and then the locally linear model-based predictive controller may be
designed. First, several vectors are defined as follows:

\[
\begin{align*}
    \hat{\mathbf{x}}(t) &= \begin{bmatrix} 
        \hat{x}(t+1|t)^T & \hat{x}(t+2|t)^T & \ldots & \hat{x}(t+N_p|t)^T
    \end{bmatrix}^T \\
    \hat{\mathbf{y}}(t) &= \begin{bmatrix}
        \hat{y}(t+1|t) & \hat{y}(t+2|t) & \ldots & \hat{y}(t+N_p|t)
    \end{bmatrix}^T \\
    \mathbf{u}(t) &= \begin{bmatrix}
        u(t) & u(t+1) & \ldots & u(t+N_c-1)
    \end{bmatrix}^T \\
    \Phi_t &= \begin{bmatrix}
        \Phi_t^T & \Phi_{t+1}^T & \ldots & \Phi_{t+N_p-1}^T
    \end{bmatrix}^T
\end{align*}
\]

(20)

where $\hat{\mathbf{x}}(t)$ is the multi-step-ahead predictive state vector, $\hat{\mathbf{y}}(t)$ is the multi-step-ahead predictive output vector, $\mathbf{u}(t)$ is the multi-step-ahead predictive control vector, $N_p$ and $N_c$ are the predictive horizon and the control horizon, respectively. Based on model (18), the multi-step-ahead predictive state and output may be given as follows:

\[
\begin{align*}
    \mathbf{x}(t) &= \mathbf{A}_t \mathbf{x}(t) + \mathbf{B}_t \mathbf{u}(t) + \mathbf{C}_t \Phi_t \\
    \mathbf{y}(t) &= \mathbf{C}_t \mathbf{x}(t) = \mathbf{C}_t \mathbf{A}_t \mathbf{x}(t) + \mathbf{C}_t \mathbf{B}_t \mathbf{u}(t) + \mathbf{C}_t \Phi_t
\end{align*}
\]

(21)

where $\mathbf{A}_t$, $\mathbf{B}_t$, $\mathbf{C}_t$ and $\mathbf{Γ}_t$ can be obtained by Eqs. (18)–(19).

\[
\begin{align*}
    \mathbf{A}_t &= \begin{bmatrix}
        \prod_{j=0}^{i} \mathbf{A}_{t+j} \\
        \prod_{j=0}^{N_p-1} \mathbf{A}_{t+j} \\
        \vdots \\
        \prod_{j=0}^{N_c-1} \mathbf{A}_{t+j}
    \end{bmatrix}, \\
    \prod_{j=i}^{k} \mathbf{A}_{t+j} &= \begin{cases}
        \mathbf{A}_{t+i} \mathbf{A}_{t+i-1} \ldots \mathbf{A}_{t+k}, & i \leq k \\
        \mathbf{I}, & i > k.
    \end{cases}
\end{align*}
\]

(22)

\[
\begin{align*}
    \mathbf{B}_t &= \begin{bmatrix}
        \prod_{j=1}^{i} \mathbf{A}_{t+j} & \mathbf{B}_t \\
        \prod_{j=1}^{N_p-1} \mathbf{A}_{t+j} & \mathbf{B}_{t+1} \\
        \vdots & \vdots \\
        \prod_{j=1}^{N_c-1} \mathbf{A}_{t+j} & \mathbf{B}_{t+N_c-2} \\
        \prod_{j=1}^{N_p-1} \mathbf{A}_{t+j} & \mathbf{B}_{t+N_c-1}
    \end{bmatrix},
\end{align*}
\]

(23)
where such sample instant for the system, and then the locally linear model-based MPC can be designed

\[
\bar{C} = \begin{bmatrix}
    c & 0 & \cdots & 0 \\
    0 & c & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & c
\end{bmatrix}_{N_p \times (n_cN_p)},
\]

(25)

\[
\Gamma_t = \begin{bmatrix}
    \prod_{j=1}^{1} A_{t+j} & \prod_{j=2}^{2} A_{t+j} & \cdots & \prod_{j=N_p}^{N_p} A_{t+j} \\
    \prod_{j=1}^{N_p-1} A_{t+j} & \prod_{j=2}^{N_p-1} A_{t+j} & \cdots & \prod_{j=N_p}^{N_p-1} A_{t+j} \\
    \vdots & \vdots & \ddots & \vdots \\
    \prod_{j=1}^{1} A_{t+j} & \prod_{j=2}^{2} A_{t+j} & \cdots & \prod_{j=N_p}^{N_p}
\end{bmatrix}
\]

(26)

The coefficient matrices \( \bar{A}_t, \bar{B}_t, \bar{C} \) and \( \Gamma_t \) depend on the working-point state prediction \( \dot{W}(t+j|t) \) at time \( t \). If the working-point state prediction is not available, one has to replace \( \dot{W}(t+j|t) \) with \( W(t) \) in computing \( \bar{A}_t, \bar{B}_t, \bar{C} \) and \( \Gamma_t \). Once the working-point is fixed at time \( t \), the coefficient matrices in (21) would be constant, which can be used as a locally linear model at such sample instant for the system, and then the locally linear model-based MPC can be designed for the system. To this end, we rewrite (21) as the following form

\[
\begin{align*}
\dot{y}(t) &= G_t \dot{u}(t) + y_0(t), \\
G_t &= \bar{C} \bar{B}_t, \\
y_0(t) &= \bar{C} \bar{A}_t x(t) + \bar{C} \Gamma_t \Phi_t,
\end{align*}
\]

(27)

Define the control increment series \( \Delta \dot{u}(t) \) and the desired output series \( \ddot{y}_r(t) \) as follows:

\[
\Delta \dot{u}(t) = [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N_c-1)]^T
\]

\[
\ddot{y}_r(t) = [y_r(t+1), y_r(t+2), \ldots, y_r(t+N_p)]^T
\]

(28)

where \( \Delta u(t) = u(t) - u(t-1) \). The objective function of the locally linear model-based MPC is

\[
\min_{\dot{u}(t)} J = \|\ddot{y}(t) - \ddot{y}_r(t)\|^2_Q + \|\dot{u}(t)\|^2_{R_1} + \|\Delta \dot{u}(t)\|^2_{R_2}
\]

s.t.

\[
\begin{align*}
y_{\text{min}} \leq \ddot{y}(t) &\leq y_{\text{max}}, \\
u_{\text{min}} \leq \dot{u}(t) &\leq u_{\text{max}}, \\
\Delta u_{\text{min}} \leq \Delta \dot{u}(t) &\leq \Delta u_{\text{max}}
\end{align*}
\]

(29)

where \( \|X\|^2_Q = X^T Q X \), \( Q, R_1 \) and \( R_2 \) are the positive definite weighting matrix of output error, \( \dot{u} \) and \( \Delta \dot{u} \), respectively. Substitute Eq. (27) into Eq. (29), and then the optimal control \( \ddot{u}(t) \) can be obtained below.

\[
\ddot{u}(t) = \arg \min_{\dot{u}(t)} J = \|G_t \dot{u}(t) + y_0(t) - \ddot{y}_r(t)\|^2_Q + \|\dot{u}(t)\|^2_{R_1} + \|\Delta \dot{u}(t)\|^2_{R_2}
\]

(30)

\[
\begin{align*}
\ddot{u}(t) &= \frac{1}{2} \dot{u}(t)^T [G_t^T Q G_t + R_1 + E^{-T} R_2 E^{-1}] \dot{u}(t) \\
&\quad + [(y_0(t)^T - \ddot{y}_r(t))^T Q G_t - u_0(t-1)^T E^{-T} R_2 E^{-1}] \dot{u}(t)
\end{align*}
\]

s.t.

\[
\begin{bmatrix}
    G_t \\
    -G_t
\end{bmatrix} \dot{u}(t) \leq \begin{bmatrix}
    y_{\text{max}} - y_0(t) \\
    -y_{\text{min}} + y_0(t)
\end{bmatrix}
\]

\[
\begin{bmatrix}
    y_{\text{min}} \leq \ddot{y}(t) &\leq y_{\text{max}}, \\
u_{\text{min}} \leq \dot{u}(t) &\leq u_{\text{max}}, \\
\Delta u_{\text{min}} \leq \Delta \dot{u}(t) &\leq \Delta u_{\text{max}}
\end{bmatrix}
\]
\[ \Delta u_{\text{min}} \leq \Delta \hat{u}(t) \leq \Delta u_{\text{max}} \] (30)

where

\[ \hat{u}(t) = \begin{bmatrix} \hat{u}(t) & \hat{u}(t+1) & \cdots & \hat{u}(t+N_c-1) \end{bmatrix}^T \]

\[ \hat{u}(t) = u_0(t-1) + E \Delta \hat{u}(t) \]

\[ u_0(t-1) = \begin{bmatrix} u(t-1) & u(t-1) & \cdots & u(t-1) \end{bmatrix}^T_{1 \times N_c} \]

\[ E = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}_{N_c \times N_c} \] (31)

The optimization of Eq. (30) is a quadratic programming (QP) problem requiring online solution. In the optimal control \( \hat{u}(t) \), the first component \( \hat{u}(t) \) is used as the real control input. Because all unknown parameters of the FWRBF-ARX model (4) have been identified offline in modeling process, the state space coefficient matrices in the optimal control (30) are known at each working-point, so the online optimization time is reduced.

The nonlinear MPC using the off-line identified neural nets-based nonlinear ARX model has been applied [19,30] and its stability has also been analyzed and verified [27]. In this work, the used neural networks in the proposed FWRBF-ARX model are not conventional RBF nets, it is a type of RBF nets with state-dependent functional weights. This improved model does not demolish the stability of the model-based MPC. As long as the optimization of Eq. (30) can find a feasible solution, the stability of the FWRBF-ARX model-based MPC will be guaranteed.

4. Real-time control to the maglev system

The FWRBF-ARX model-based MPC (FWRBF-ARX-MPC) designed in Section 3 is used to control the maglev system, and the model orders in (4) and the dimension of the state in (17) are determined as \( n_y = 7 \), \( n_u = 6 \) and \( k_n = 7 \) for the system. To implement the FWRBF-ARX-MPC for the maglev system, a MPC control system under the MATLAB/SIMULINK 2010b environment is designed to control the steel ball levitation. The sampling period is set as 5 ms. The platform of the control system is shown in Fig. 8. The module ‘Real Control’ is the real-time module of the controlled system in Fig. 2 and is the same with the module shown in Fig. 3, it achieves the input signal sampling and the A/D or D/A converter of signals. The ‘FWRBF_MPC’ is an S-Function written in C language and completes the MPC computation for the maglev system. \( r_1 \), \( r_2 \) and \( q \) are the factors of the scalar weights

\[ R_1 = r_1 I_{N_y \times N_y}, \quad R_2 = r_2 I_{N_y \times N_y}, \quad Q = q I_{N_y \times N_y} \] in Eq. (30), where \( I \) denotes the unit matrix. \( Wt \) that is the first term of \( W(t) \) in model (4) denotes the working-point state, ‘yr’ that is \( y_r(t+i) \) in Eqs. (28)–(30) is the desired output, \( y_1 \) is the system output at the current time, where we select \( y_1 \) as the working-point state \( Wt \). The ‘Gain’ is used to convert the output voltage value (unit: V) into the position of the ball (unit: mm), which is obtained by calibrating the photoelectric sensor in Fig. 2.

Note that for the fast-responding maglev system, the computation time of the QP optimization using the function ‘QUADPROG’ based on the interior-reflective Newton method in the MATLAB Optimization Toolbox may exceed the sampling period 5 ms. Therefore, we use C language Optimization function ‘QUADPROG’ based on the Goldfarb-Idnani method [31] to solve the quadratic program problem (30), which may improve the computation efficiency. The computation cost of QP problem (30) will be analyzed in Section 5 in detail.
In this section, the designed predictive controller is applied to control the levitation movement of the steel ball. In order to evaluate the control performance of the FWRBF-ARX-MPC, three other control approaches which are conventional PID control in Fig. 3, the RBF-ARX model-based MPC (RBF-ARX-MPC) [19] and the linear ARX model-based MPC (ARX-MPC) with the same MPC structure as the FWRBF-ARX-MPC, are also applied into the maglev system for comparison. Moreover, the experiments tracking rectangular wave, which can test the dynamic response capability of the control system, are done to exhibit the control results of different control strategies. These rectangular waves are set around -9 mm position which is at the middle position of the operational range of the ball. To get a fair comparison, the final controller parameters used in the four control approaches are attained after trial and error, which are the well-tuned values to make the ball track the reference signal well and are presented in Table 3.

Note that because the obtained linear ARX model in Section 2.4 is identified using a set of global observation data in Fig. 5, this ARX model-based MPC may not be able to achieve stable control to the steel ball. It is necessary to use a set of local observation data to identify a new linear ARX model for getting better control. The new observation data is generated under the control of PID in Fig. 3 according to the given reference trajectory \( y_r = 0.56 \sin t - 9 \) (mm). The identified model order is \( n_y = 6, n_u = 2 \). In the control process, the modeling errors are compensated into the bias \( \phi_0 \) of three models.

Table 3
Parameters of four controllers.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>( K_p = 1.4, K_i = 0.04, K_d = 18 )</td>
</tr>
<tr>
<td>ARX-MPC</td>
<td>( N_p = 9, N_c = 4, R_1 = 0.03, R_2 = 0.55, Q = 2.25 )</td>
</tr>
<tr>
<td>RBF-ARX-MPC</td>
<td>( N_p = 12, N_c = 10, R_1 = 0.0003, R_2 = 0.23, Q = 1.8 )</td>
</tr>
<tr>
<td>FWRBF-ARX-MPC</td>
<td>( N_p = 12, N_c = 4, R_1 = 0.0002, R_2 = 0.16, Q = 1.8 )</td>
</tr>
</tbody>
</table>

In this section, the designed predictive controller is applied to control the levitation movement of the steel ball. In order to evaluate the control performance of the FWRBF-ARX-MPC, three other control approaches which are conventional PID control in Fig. 3, the RBF-ARX model-based MPC (RBF-ARX-MPC) [19] and the linear ARX model-based MPC (ARX-MPC) with the same MPC structure as the FWRBF-ARX-MPC, are also applied into the maglev system for comparison. Moreover, the experiments tracking rectangular wave, which can test the dynamic response capability of the control system, are done to exhibit the control results of different control strategies. These rectangular waves are set around -9 mm position which is at the middle position of the operational range of the ball. To get a fair comparison, the final controller parameters used in the four control approaches are attained after trial and error, which are the well-tuned values to make the ball track the reference signal well and are presented in Table 3.

Note that because the obtained linear ARX model in Section 2.4 is identified using a set of global observation data in Fig. 5, this ARX model-based MPC may not be able to achieve stable control to the steel ball. It is necessary to use a set of local observation data to identify a new linear ARX model for getting better control. The new observation data is generated under the control of PID in Fig. 3 according to the given reference trajectory \( y_r = 0.56 \sin t - 9 \) (mm). The identified model order is \( n_y = 6, n_u = 2 \). In the control process, the modeling errors are compensated into the bias \( \phi_0 \) of three models.
The step response control results of the ball under the control of four controllers are shown in Figs. 9–14, in which the first subfigure is the system output, i.e. the position of the ball, and the second subfigure is the control voltage. In all figures, ‘yr’ that is the same as ‘yr’ in Fig. 8 denotes the given reference signal $y_r$ in Eqs. (28)–(30). The step amplitudes change from 1.125 mm to 7.875 mm. Figs. 9–11 contain the experiment results of the PID control and three different ARX model-based MPCs. In order to show more subtle differences, some parts of the step responses in Figs. 9–10 are locally magnified. It may be seen that the overshoot of PID control is the largest and is much larger than that of two global nonlinear models (FWRBF-ARX model and RBF-ARX model)-based MPCs. The FWRBF-ARX-MPC can make the ball faster go into the stable state compared to other controllers. When the step amplitude gradually increases, the increment of overshoot using the FWRBF-ARX-MPC is the smallest among four controllers, but the overshoot of PID control and ARX-MPC are very large. This may result in that the PID control and the ARX-MPC are not able to control the ball to track a given step signal when the jump amplitude is increased to a certain value. As shown in Fig. 11, the ball is out of control and falls down under the control of PID when the negative step amplitude is 3.825 mm. In Fig. 12, the ball is out of control under the control of ARX-MPC when the step amplitude becomes 4.500 mm. Therefore, the control range of PID controller is less than 3.825 mm and that of the ARX-MPC is less than 4.5 mm. These results are attributed to the maglev system’s strong nonlinearity which restricts PID control range, and due to that the linear ARX model can only describe the system’s dynamics in a small range. The FWRBF-ARX-MPC and RBF-ARX-MPC are based on a global approximate nonlinear model and may control the ball to track a large step signal, as shown in Figs. 12–14.

Fig. 9. Step response of four controllers to 1.125 mm jump amplitude.
Fig. 10. Step response of four controllers to 2.250 mm jump amplitude.

Fig. 11. Step response of four controllers to 3.825 mm jump amplitude.
Fig. 12. Step response of three MPCs to 4.500 mm jump amplitude.

Fig. 13. Step response of FWRBF-ARX-MPC and RBF-ARX to 6.750 mm jump amplitude.
With the increase of step amplitude, the ball is still able to track stably a given step signal under the control of FWRBF-ARX-MPC and RBF-ARX-MPC, it shows good global control ability of the two models-based MPCs. Under the control of the two controllers, their rising times and overshoots in the rising stage are almost the same. However, the overshoot of RBF-ARX-MPC in the negative step stage is larger than that of FWRBF-ARX-MPC and reaches to 18.222% in Fig. 13 and 23.683% in Fig. 14. The adjustment time of FWRBF-ARX-MPC is slightly smaller than that of RBF-ARX-MPC, it makes the ball more quickly reach the steady state. Dynamic performances of step response to four controllers are computed and presented in Table 4, where $\sigma_p$ is the overshoot and $t_r$ is the adjustment time, ‘up’ represents the rising step stage, and ‘down’ denotes the negative step, ‘-’ denotes out of control. From Table 4, one can see that the control performance of the proposed FWRBF-ARX-MPC is much better than that of the other controllers.

As a whole, the control range of conventional PID and linear ARX model-based predictive control is small due to the local property of the controllers. The FWRBF-ARX-MPC and RBF-ARX-MPC are all global control strategy and are able to control the ball movement in a wide range. The results in Figs. 9–14 show that the control performances are different at the rising stage and the negative step. The main reason for this phenomenon may be that the magnetic field generated by the electromagnet in Fig. 2 is strongly nonlinear and asymmetric. The magnetic lines are dense and a small move may result in a large change of the electromagnetic force when the ball is close to the electromagnet; while the magnetic lines are sparse and the nonlinearity is stronger when the ball is at greater distance away from the electromagnet. The FWRBF-ARX model contains more local information of the working-point, so that the control is more effective when the ball is closer to /farther away from the electromagnet. Compared with the RBF-ARX-
MPC, the control performance of the FWRBF-ARX-MPC in the negative step is better and more stable when the jump amplitude is larger, while the positive step responses of two controllers are approximate. It verifies that the proposed FWRBF-ARX modeling method may better represent the global behavior of the maglev system. The response to a globally changed signal under the control of the FWRBF-ARX-MPC is shown in Fig. 15, and one can see that the ball is able to track the reference trajectory quite well, and there exists an overshoot if the ball is very close or far away from the electromagnet and the changing is rapid. This is accordant with the results exhibited in Figs. 9–14.

5. Computational effort

The computational complexity of the FWRBF-ARX-MPC is discussed in this section. Computation time of QP problem in predictive control is mainly affected by the number of input and output variables, control horizon and predictive horizon. In general, the longer the control horizon is, the better the control performance is, but the larger the optimization computational effort will be. Zou et al. [32] and Wei et al. [33] computed the optimization time of predictive control by changing the number of input/output and control horizon, and concluded that the complexity of optimization computation with constraints (30) is proportional to the third power of product of the number of control variables and the length of control horizon, i.e. \( O(nN_c)^3 \), where \( n \) is the number of control variables and \( N_c \) is the control horizon. For the maglev system, if \( n \) and \( N_c \) is the same, the computation complexity of the FWRBF-ARX-MPC is almost the same as that of the RBF-ARX-MPC when solving QP problem (30). On the other hand, it is inevitable to compute the matrix \( H = G_r^T Q G_r + R_1 + E^{-T} R_2 E^{-1} \) and \( F = (y_0(t)^T - \hat{y}_r(t)^T) Q G_r - u_0(t-1)^T E^{-T} R_2 E^{-1} \) in (30) when online optimizing the control variable \( \bar{u}(t) \). This is the main factor affecting the speed of solving the optimization problem (30). The dimensions of \( H \) and \( F \) both are \( nN_c \times nN_c \) and \( n \times nN_c \) in the FWRBF-ARX-MPC and RBF-ARX-MPC, respectively. The computational complexity of QP problem (30) is approximately
Therefore, one can see that although the FWRBF-ARX model structure is changed on the basis of the RBF-ARX model, the computation complexity for the FWRBF-ARX-MPC is not increased. However, the FWRBF-ARX-MPC must compute the linear weights depending on the working-point state in RBF networks, so the FWRBF-ARX-MPC needs to spend a little more time than the RBF-ARX-MPC to calculate the matrix $H$ and $F$.

The comparisons of computing time for the two MPCs are carried out in the simulating experiments using ‘QUADPROG’ in MATLAB toolbox and ‘QUADPROG’ in C language, respectively. The FWRBF-ARX-MPC and RBF-ARX-MPC programs with the same structure are run under the MATLAB/SIMULINK 2010b environment. The simulation block is similar to the real-time block in Fig. 8, and the controlled object is represented by the identified FWRBF-ARX model and RBF-ARX model, respectively. The simulation step responses to 6.75 mm jump amplitude are shown in Fig. 16. Moreover, the computation costs of the QP problem (30) with different optimization programs are obtained and compared, which are shown in Fig. 17 and Table 5. The results show that the computation time of the optimization problem using Matlab program in each case is much longer than that of using C routine, and the former even may be more than 5 ms in some cases. However, the sample period is just 5 ms in our case, so the optimization routine using C language is used to get the optimal control $\tilde{u}(t)$ in our real-time experiments. On the other hand, one can see in Table 5 that the average computation time of the FWRBF-ARX-MPC is only slightly larger than that of the RBF-ARX-MPC; this is coincident with the conclusion that the two control strategies with the same control horizon have the same computational complexity. It is confirmed that the proposed FWRBF-ARX-MPC can improve the control performance of the maglev system but do not increase the computation complexity.
Fig. 16. Simulation results of FWRBF-ARX-MPC and RBF-ARX-MPC to step signal (amplitude: 6.75 mm). (a) FWRBF-ARX-MPC. (b) RBF-ARX-MPC.
Fig. 17. Computation time of FWRBF-ARX-MPC and RBF-ARX-MPC to step signal (amplitude: 6.75 mm). (a) FWRBF-ARX-MPC. (b) RBF-ARX-MPC.

Table 5
Computation time of two predictive controllers.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Computation time using ‘QUADPROG’ [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MATLAB</td>
</tr>
<tr>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>RBF-ARX-MPC</td>
<td>2.352</td>
</tr>
<tr>
<td>FWRBF-ARX-MPC</td>
<td>2.759</td>
</tr>
</tbody>
</table>
6. Conclusions

In this work, a linear function depending on system working-points state was used as the output weight of RBF networks and this modified RBF networks were used to approximate the function type regressive weights of nonlinear ARX model, which was referred to as FWRBF-ARX model. We proposed the identification approach to the model, and built a FWRBF-ARX model to characterize a nonlinear maglev system. This model possesses locally linearity at each working-point and globally nonlinearity in whole working range. This is a data-driving modeling method for a class of nonlinear system, and is a good choice especially for systems whose physical model and/or parameters are not easily acquired. For capturing and quantifying dynamics of an unstable, fast-response maglev system, based on the maglev system's actual historical input/output data, the parameters and structure of a FWRBF-ARX model were estimated offline using the SNPOM. The identified results showed that the FWRBF-ARX model has smaller one-step-ahead predictive errors, smaller AIC value and smaller long-term predictive errors than that of the estimated RBF-ARX and linear ARX model, and the former can represent the dynamics of the unstable, fast-response maglev system very well. This verified the effectiveness and feasibility of the FWRBF-ARX modeling for the maglev system. Moreover, the identified FWRBF-ARX model-based predictive controller was designed to achieve the stable levitation of the steel ball in the electromagnetic field. The real-time control results to the maglev system illustrated that the FWRBF-ARX-MPC exhibited much better control performance than that of PID control, RBF-ARX-MPC and linear ARX-MPC, and it did not increase the computational complexity. The real-time control application results demonstrated that the proposed FWRBF-ARX modeling technique and the model-based predictive control approach are suitable for controlling such an unstable, fast-response maglev system.

The MPCs used in the paper are linear predictive controls based on a local linearization model at certain working-point of the system. They did not use the future working-points state information. In fact, it can be seen from model (4) that the state-dependent ARX model can be used for predicting the future output based on the present state and historical knowledge, but the MPC using the complete future prediction will need much heavier computation burden than that of the presented MPCs in this paper. Our future work is to investigate how to overcome this drawback.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (71271215, 71221061 and 61403045), the International Science & Technology Cooperation Program of China (2011DFA10440) and the Collaborative Innovation Center of Resource-conserving & Environment-friendly Society and Ecological Civilization of China. The authors would like to thank the editors and referees for their valuable comments and suggestions that largely improved the quality of this paper.

References


