Research Article

Data-Based Control for Humanoid Robots Using Support Vector Regression, Fuzzy Logic, and Cubature Kalman Filter

Liyang Wang,¹ Ming Chen,¹ Gai Li,¹ and Yongqing Fan²

¹Department of Electronic Engineering, Shunde Polytechnic, Foshan, Guangdong 528300, China

²School of Automation, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Liyang Wang; ddd0wwl@sohu.com

Received 28 March 2016; Accepted 29 May 2016

Academic Editor: Mohammad D. Aliyu

Copyright © 2016 Liyang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Time-varying external disturbances cause instability of humanoid robots or even tip robots over. In this work, a trapezoidal fuzzy least squares support vector regression- (TF-LSSVR-) based control system is proposed to learn the external disturbances and increase the zero-moment-point (ZMP) stability margin of humanoid robots. First, the humanoid states and the corresponding control torques of the joints for training the controller are collected by implementing simulation experiments. Secondly, a TF-LSSVR with a time-related trapezoidal fuzzy membership function (TFMF) is proposed to train the controller using the simulated data. Thirdly, the parameters of the proposed TF-LSSVR are updated using a cubature Kalman filter (CKF). Simulation results are provided. The proposed method is shown to be effective in learning and adapting occasional external disturbances and ensuring the stability margin of the robot.

1. Introduction

In general, lots of experiences are needed to turn the parameters of the controller for humanoid walking robots [1, 2]. At the same time, the turned parameters could be out of operation once external disturbances occurred [3, 4]. It is still a giant challenge for humanoid robots to walk autonomously in disturbed environments. How to improve the antidisturbance ability of humanoid robots using the data online and offline is an interesting problem to be settled.

Traditionally, an accurate dynamic model of the considered robot should be built in order to implement the desired high-quality control [5, 6]. The dynamic model of humanoid robots is a set of strong coupling nonlinear ordinary differential equations about the joint variables. When humanoid robots walk slowly, the coupling among the joints can be treated as disturbances. In this case, proportion integration differentiation (PID) control for each independent joint can be adopted. Additionally, inverse dynamics feed-forward control, decoupling control, and feedback linearization control also can be considered. All the above-mentioned methods have a common feature that they are dependent on the established mathematical model strongly.

The control effect could be great if the system model is known exactly. However, these methods need to be improved in practical applications because there always exist external disturbances and the system model cannot be known exactly [7, 8]. Due to the cumulative errors in disturbances, when the robot is tracking the walking pattern planned in advance, the difference between actual motion states and target values will increase rapidly. As a consequence, humanoid robots always fall down after a few steps. To keep stable and sustained humanoid walking, the control systems must be improved to cope with the disturbances.

To realize the stable walking of humanoid robots, researchers proposed some effective methods, such as stability controls based on linear inverted pendulum model, stability controls based on ZMP theory, and attitude controls for the upper part of robot. By summarizing the existing humanoid prototypes, ZMP-based methods are the most popular and practical [9–13]. When the humanoid dynamics (the masses of each module, the center of mass, and the moment of inertia) is known exactly, the gait of a humanoid robot can be obtained by solving the ZMP equation. However, it is really difficult to collect all the precise parameters from a physical robot. As a result, several simplified methods are

proposed to guarantee the stability criterion of ZMP. For example, Liu et al. designed an effective fuzzy logic (FL) controller for humanoid walking robots using ZMP as one of the antecedents [14]. Inspired by this literature, we try to realize a kind of data-based control with implicit constraints of ZMP.

Considering the time-varying external disturbances, it is necessary to deduce the control torques referring to the time-varying states of the humanoid joints. The support vector machine (SVM) is supervised learning methods with associated learning algorithms that analyze data [15–17]. SVMs can be used to solve the classification problems [18, 19] and also the regression problems [20–22]. In this paper, support vector regression (SVR) is used to express the scene of solving the regression problems using SVM. The formulation of SVM employs the structural risk minimization (SRM) principle, which has been shown to be superior to the empirical risk minimization (ERM) based on infinite samples. SRM is a technique where nested sets of functions of different complexity, controlled by the regularization term, are considered. One could select then the one which is minimizing the upper bound on the generalization error. This feature makes SVM more efficient in resolving the learning problems with limited training data. To cope with time-varying external disturbances, a time-based fuzzy SVR will be proposed to learn the dynamics between the states and the control torques of each joint in this work.

On the other hand, the effectiveness of time-based fuzzy SVR depends on the design of the SVR and the parameters of the fuzzy system. Kalman filter has been used in algorithm studies on the training of neural networks and fuzzy systems. Singhal and Wu [23] demonstrated that the EKF could serve as the basis for training MLP networks by treating the weights of the network as a nonlinear dynamical system. Inspired by the successful application of the Kalman filter for training neural networks [24] and for defuzzification strategies [25], Simon [26] built a nonlinear system to train fuzzy systems using an extended Kalman filter. Recently, there is a research [27] that embedded a third-degree spherical-radial cubature rule into the Bayesian filter to build a kind of new filter named CKF. CKF has demonstrated excellent performance in solving nonlinear filtering problems with minimal computational effort [28–31]. Therefore, we will explore approaches to train the proposed time-based fuzzy SVR using CKF.

The contributions of this work can be summarized as follows:

- (i) For the first time, a trapezoidal fuzzy SVR is proposed to cope with time-varying external disturbances imposed on humanoid robots.
- (ii) For the first time, a novel approach for training the trapezoidal fuzzy SVR using CKF is presented.

The organization of this paper is as follows. In Section 2, the backgrounds of humanoid robots, SVR, and CKF are presented. The details for proposed framework are presented in Section 3. Simulation results are provided in Section 4, followed by the conclusions in Section 5.

2. Background

2.1. Dynamic Balance of Humanoid Robots. The dynamic equations of the single support phase (SSP) can be written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau + \tau_d, \quad (1)$$

where $\theta \in R^n$ is the generalized coordinate and $M(\theta) \in R^{n \times n}$ is the inertia matrix. $C(\theta, \dot{\theta}) \in R^{n \times n}$ denotes the matrix of centripetal acceleration and Coriolis terms, $G(\theta) \in R^{n \times n}$ is the gravity vector, and $\tau \in R^n$ denotes the input torque vector during the SSP. The external disturbances are represented by $\tau_d \in R^n$.

The dynamic equations of the double support phase (DSP) can be written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + J^T(\theta)F = \tau + \tau_d, \quad (2)$$

where $J^T(\theta) \in R^{l \times n}$ is a Jacobian matrix and $F \in R^l$ is the force vector of constraints caused by the contact with ground.

To analyze the stability of the humanoid motions, the ZMP theory is used as the criterion of dynamic humanoid balance in this work. The concept of zero moment point (ZMP) has been applied to many famous humanoid robots successfully, such as ASIMO [11] of Honda. ZMP is a well-known concept introduced in 1990 [32], which is a point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes. At a given time instant, dynamic balance of legged systems is ensured if the ZMP is inside the support area.

2.2. Support Vector Regression. In this section, we briefly review the basis of the theory of SVR.

Given a labeled training data set $\{x_i, y_i\}$ ($i = 1, 2, \dots, N$), $x_i \in R^n$ is the input vector of the system and $y_i \in R$ is the output vector. The basic idea of SVR is mapping the input space into a higher dimension feature space using the non-linear mapping function and searching the optimal linear regression function in this feature space. Objective function of the least squares SVR (LS-SVR) [17] is

$$\begin{aligned} \min \quad & \frac{1}{2}\omega^T\omega + \frac{1}{2}C\sum_{i=1}^N\xi_i^2 \\ \text{s.t.} \quad & y_i = \omega^T\phi(x_i) + b + \xi_i, \quad i = 1, \dots, N, \end{aligned} \quad (3)$$

where C is a positive real constant for tuning. The error of the regression becomes smaller when the value of C is smaller. ω is a weight vector and b is a bias; ξ_i is the positive slack variable enabling dealing with permitted errors. To solve this optimization problem we construct the Lagrangian

$$\begin{aligned} L(\omega, b, \xi, \alpha) \\ = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^N\xi_i^2 - \sum_{i=1}^N\alpha_i[\omega\phi(x_i) + b + \xi_i - y_i], \end{aligned} \quad (4)$$

and we find the saddle point of $L(\omega, b, \xi, \alpha)$, where $x_i \in R^n$ is the input vector of the system and $y_i \in R$ is the output vector. $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$ is a vector of the Lagrange multipliers. The parameters must satisfy the following conditions:

$$\frac{\partial L}{\partial \omega} = 0 \longrightarrow$$

$$\omega = \sum_{i=1}^N\alpha_i\phi(x_i),$$

$$\frac{\partial L}{\partial b} = 0 \longrightarrow$$

$$\begin{aligned}
\sum_{i=1}^N \alpha_i &= 0, \\
\frac{\partial L}{\partial \xi} &= 0 \longrightarrow \\
\alpha_i &= C\xi_i, \\
\frac{\partial L}{\partial \alpha} &= 0 \longrightarrow \\
\omega\phi(x_i) + b + \xi_i - y_i &= 0.
\end{aligned} \tag{5}$$

Eliminating ω and ξ , problem (3) can be transformed into

$$\begin{bmatrix} 0 & A^T \\ A & \Omega + C^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}. \tag{6}$$

where $y = [y_1, y_2, \dots, y_N]^T$, $A = [1, 1, \dots, 1]^T$, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T$, Ω is a square matrix, which has elements of $\Omega_i = K(x_i, x) = \phi(x_i)^T \phi(x)$, and $K(x_i, x)$ is a kernel function. For example, $K(x_i, x) = \exp[-(x_i - x)^2/2\sigma^2]$ denotes a radial basis function (RBF) kernel. Submitting the optimal α and b , the regression function of the least squares SVR [17] is

$$y(x) = \sum_{i=1}^N \alpha_i K(x_i, x) + b. \tag{7}$$

2.3. Cubature Kalman Filter. To describe the CKF, we consider the filtering problem of a nonlinear dynamic system with additive noise, whose state space model is defined by a process equation and a measurement equation in discrete time:

$$\begin{aligned}
x_k &= f(x_{k-1}, u_{k-1}) + w_{k-1}, \\
z_k &= g(x_k, u_k) + v_k,
\end{aligned} \tag{8}$$

where x_k is the state of the dynamic system at discrete time k ; $f(\cdot)$ and $g(\cdot)$ are some known functions; u_k is the control input; z_k is the measurement; $\{w_{k-1}\}$ and $\{v_k\}$ are independent process and measurement Gaussian noise sequences with zero means and covariances Q_{k-1} and R_k , respectively.

When we deal with a problem of state estimation using a nonlinear filtering, the integrals for the means and variances of the states can be expressed as the form of a Gaussian-weighted integral. Consider a Gaussian-weighted integral of the following form:

$$I(f) = \int_{U^n} f(x) e^{-x^T x} dx, \tag{9}$$

where $f(x)$ is an arbitrary function and U^n is the region of integration. There are different integration methods to derive different nonlinear filters.

For a CKF, the spherical-radial cubature rule is adopted to implement the integration. Let $x = rt(t^T t = 1, r \in [0, +\infty))$,

and integration (9) can be separated into radial integration and spherical integration. That is,

$$R = \int_0^\infty S(r) r^{n-1} e^{-r^2} dr, \tag{10}$$

$$S(r) = \int_{S_n} f(rt) d\rho(t),$$

where S_n is the surface of the sphere defined by $\rho \in S_n$. Using Lagrange integration, the radial integration can be rewritten as

$$R \approx \frac{1}{2} \Pi\left(\frac{n}{2}\right) S\left(\sqrt{\frac{n}{2}}\right), \tag{11}$$

where $\Pi(n) = \int_0^\infty x^{n-1} e^{-x} dx$. Using a cubature rule of degree three, the spherical integration can be rewritten as

$$S(r) \approx \sum_{i=1}^{2n} \frac{\sqrt{\pi^n}}{n \cdot \Pi(n/2)} f(\langle 1 \rangle_i), \tag{12}$$

where $\langle 1 \rangle_i$ denotes the i th column of set $\langle 1 \rangle$. For example, $\langle 1 \rangle = (\langle 1, 0 \rangle^T, \langle -1, 0 \rangle^T, \langle 0, -1 \rangle^T, \langle 0, 1 \rangle^T)$ when $n = 2$.

Combining (11) and (12), the spherical-radial cubature rule is as follows [27]:

$$I(f) \approx \frac{\sqrt{\pi^n}}{2n} \sum_{i=1}^{2n} f\left(\sqrt{\frac{n}{2}} \langle 1 \rangle_i\right). \tag{13}$$

For standard Gaussian distribution, (13) can be rewritten as

$$\begin{aligned}
I_N(f) &\approx \int_{R^n} f(x) N(x; 0, I) dx \\
&= \frac{1}{\sqrt{\pi^n}} \int_{R^n} f(\sqrt{2}x) e^{-x^T x} dx,
\end{aligned} \tag{14}$$

where $N(x; 0, I)$ denotes the Gaussian density of x with mean 0 and covariance I . Combining (9) and (14), we get

$$I_N(f) \approx \frac{1}{2n} \sum_{i=1}^{2n} f\left(\sqrt{\frac{2n}{2}} \langle 1 \rangle_i\right) = \sum_{i=1}^{2n} \beta_i f(\eta_i), \tag{15}$$

where n is the dimension of the state vector. The point (η_i, β_i) is called cubature point here. η_i and β_i can be calculated as follows:

$$\begin{aligned}
\eta_i &= \sqrt{\frac{2n}{2}} \langle 1 \rangle_i, \\
\beta_i &= \frac{1}{2n},
\end{aligned} \tag{16}$$

$$i = 1, 2, \dots, 2n.$$

This means that, for the third-degree spherical-radial rule, it entails a total of $2n$ cubature points. After calculating the cubature points, we use the cubature-point set to numerically compute the integrals and obtain the CKF algorithm; details of time update and measurement update can be found in the literature [27].

FIGURE 1: Data-based control for humanoid walking robots using support vector regression, fuzzy logic, and cubature Kalman filter.

FIGURE 2: The trapezoidal fuzzy membership function for humanoid walking samples.

3. Data-Based Control for Humanoid Walking Robots

External disturbance is one of the key issues which influence the stability of humanoid walking robots. On the other side, it is difficult to measure the external disturbances directly. Based on these facts, we turn away to focus attention on the data of the humanoid states and the corresponding control torques because the pattern of the disturbances can be disclosed using the varying data collected from the humanoid robots to some extent.

First of all, the states and the control torques of the humanoid joints are collected from a simulated stable walking robot. Then, a data-based controller considering the varying states of the humanoid robot is designed using SVR and fuzzy theory. To optimize the controller, a CKF is designed to train the parameters of the SVR and the fuzzy system. The complete framework for data-based humanoid control using SVR, FL, and CKF is shown in Figure 1.

FIGURE 3: Simplified model of the simulated humanoid robot.

3.1. Data Collecting from the Simulations. Data for training the controller is collected by implementing simulation experiments. Two kinds of data are collected from the simulated humanoid robot, including the joint angles and the driving torques. The way we get the training data is described in detail next.

First of all, data of the joint angles are generated from reference trajectory planned offline. The trajectories can be represented as follows:

$$x_a(i) = \left(\frac{s}{\pi}\right) \left[\frac{2\pi i}{N} - \sin\left(\frac{2\pi i}{N}\right) \right],$$

$$z_a(i) = \left(\frac{d}{\pi}\right) \left[1 - \cos\left(\frac{2\pi i}{N}\right) \right],$$

FIGURE 4: Learning results for Submodel 1 of the SSP (the step length is 0.16 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

$$\begin{aligned} x_h(i) &= \frac{x_a(i)}{2} + \frac{s}{2}, \\ z_h(i) &= \frac{z_a(i)}{2} + l_{th} + l_{sh} - \frac{d}{2}, \end{aligned} \quad (17)$$

where x_h , z_h represent the position of the hip and x_a , z_a represent the position of the swinging ankle joint. s denotes the walking step length, and d denotes the height of swinging ankle. N denotes the total number of samples for a step, i denotes the index of the samples, and l_{th} , l_{sh} represent the length of lower limbs.

Secondly, a proportion integration differentiation (PID) controller is used to obtain the driving torques. In this work,

the initial driving torques of all the joints are obtained using this PID controller. Then the key driving torques, including driving torques for the support hip and support ankle in the SSP and driving torques for the knees in the DSP, are improved using SVR, FL, and CKF. The PID controller is as follows:

$$\tau_j = P_j e_j + I_j \int_0^T e_j dt + \frac{D_j de_j}{dt}, \quad (18)$$

where τ_j ($j = 1, \dots, n$) is the torque of the joints. e_j denotes the offset of the desired reference trajectories and the actual trajectories. T is the integral period. The proportional gains P_j , integral gains I_j , and differential gains D_j are slightly modified by the trial-error method.

FIGURE 5: Learning results for Submodel 1 of the SSP (the step length is 0.18 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

3.2. Designing the Controller Using the Collected Data. A TF-LSSVR is proposed in this section to design the controller using the collected data.

3.2.1. Humanoid Controller to Be Built Using the TF-LSSVR. Based on the existing literature [14], when the ZMP criterion is satisfied, the dynamics between the torque control inputs and the joint angles of the DSP can be presented as

$$\begin{aligned}\tau_{\text{left_knee}} &= h(\theta_{\text{left_knee}}, \theta_{\text{right_knee}}), \\ \tau_{\text{right_knee}} &= l(\theta_{\text{left_knee}}, \theta_{\text{right_knee}}),\end{aligned}\quad (19)$$

where $\tau_{\text{left_knee}}$ and $\tau_{\text{right_knee}}$ are the driven torques of the left knee and the right knee. $\theta_{\text{left_knee}}$ and $\theta_{\text{right_knee}}$ are the joint

angles of the left knee and the right knee. $h(\cdot)$ and $l(\cdot)$ are the nonlinear dynamics in the DSP to be learned using the proposed TF-LSSVR.

On the other side, when the ZMP criterion is satisfied, the dynamics between the torque control inputs and the joint angles of the SSP can be presented as [14]

$$\begin{aligned}\tau_{\text{sup_hip}} &= f(\theta_{\text{sup_hip}}, \theta_{\text{sup_ankle}}), \\ \tau_{\text{sup_ankle}} &= g(\theta_{\text{sup_hip}}, \theta_{\text{sup_ankle}}),\end{aligned}\quad (20)$$

where $\tau_{\text{sup_hip}}$ and $\tau_{\text{sup_ankle}}$ are the driven torques of the supporting hip and the supporting ankle. $\theta_{\text{sup_hip}}$ and $\theta_{\text{sup_ankle}}$ are the joint angles of the supporting hip and the supporting

FIGURE 6: Learning results for Submodel 1 of the SSP (the step length is 0.20 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

ankle. $f(\cdot)$ and $g(\cdot)$ are the nonlinear dynamics in the SSP to be learned using the proposed TF-LSSVR.

For illustration purposes, we will expound the situation in the SSP mainly. The solution for the DSP can be easily deduced in the same way. Then, the humanoid controller to be built can be simply presented as follows:

$$\tau = G(\Theta), \quad (21)$$

where $G(\cdot)$ is the nonlinear dynamics that the TF-LSSVR tries to build. τ is the driving torque, and $\Theta = (\theta_{\text{sup_hip}}, \theta_{\text{sup_ankle}})$ is a vector of joint angles.

3.2.2. Objective Functions of the TF-LSSVR. When the timeliness of the training data is considered, the collected data from current steps are “more important,” and those from past steps are “less important.” Based on this, in the proposed

objective functions of the TF-LSSVR, a TFMF is used to design learning weights of the collected data. Taking the learning of driving torque of the supporting hip as an example and designing a learning weight r_k for each of the training samples, the training sample set can be denoted by $\{\theta_{\text{sup_hip}}^{(k)}, \theta_{\text{sup_ankle}}^{(k)}, \tau_{\text{sup_hip}}^{(k)}, r_k\}$, $0 \leq r_k \leq 1$, $k = 1, 2, \dots, N$. Then the objective function of the TF-LSSVR for training the driving torque of the supporting hip is

$$\begin{aligned} \min \quad & \frac{1}{2} \omega^T \omega + \frac{1}{2} C r_k \sum_{k=1}^N \xi_k^2 \\ \text{s.t.} \quad & \tau_{\text{sup_hip}}^{(k)} \\ & = \omega^T \phi(\theta_{\text{sup_hip}}^{(k)}, \theta_{\text{sup_ankle}}^{(k)}) + b_{\text{sup_hip}} + \xi_k, \\ & k = 1, \dots, N, \end{aligned} \quad (22)$$

FIGURE 7: Learning results for Submodel 2 of the SSP (the step length is 0.16 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

where ω is a weight vector. $\phi(\cdot)$ is a nonlinear mapping function. C is a penalty coefficient, ξ_k is a positive slack variable, and $b_{\text{sup_hip}}$ is the corresponding bias. $\{\theta_{\text{sup_hip}}^{(k)}, \theta_{\text{sup_ankle}}^{(k)}, \tau_k^{(k)}, r_k\}$ is the k th sample, and N is the number of the samples. The r_k is the fuzzy learning weight for the training samples, which can be calculated using the fuzzy membership functions designed in the next section. It is noted that in literature [17] also a weighting scheme similar to (22) has been proposed but it is based on robust statistics. In this paper, a different weighting scheme is proposed.

3.2.3. Designing the Learning Weights Using TFMF. The walking data from different time can be evaluated using some linguistic terms such as “more important” or “less

important.” When the timeliness of the training data is considered, the collected data from current steps are “more important,” and those from past steps are “less important.”

For this reason, a TFMF, which is the left half of a trapezoidal function, is proposed as the membership function of humanoid walking samples. The proposed time-related trapezoidal fuzzy membership function is shown in Figure 2. The formula for the trapezoidal fuzzy membership function is as follows:

$$r_k = \begin{cases} \frac{1-h}{t_{k-m}-t_{k-m-n}}t + \frac{t_{k-m}h-t_{k-m-n}}{t_{k-m}-t_{k-m-n}}, & t \in [t_{k-m-n}, t_{k-m}), \\ 1, & t \in [t_{k-m}, t_k], \end{cases} \quad (23)$$

FIGURE 8: Learning results for Submodel 2 of the SSP (the step length is 0.18 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

where r_k is the trapezoidal fuzzy membership function of time t_k ($k = 1, 2, \dots, N$), $r_k \in [h, 1]$, and $h \in (0, 1)$ is a tuning parameter. t_{k-m-n} and t_k are the beginning and the ending of the time window, respectively, and t_{k-m} is a time point between the beginning and the ending.

From Figure 2 and (23), it can be seen that the time window $[t_{k-m-n}, t_k]$ is divided into two parts. The first part $[t_{k-m}, t_k]$ contains the newest walking samples, which are designed to have a full fuzzy membership grade. The second part $[t_{k-m-n}, t_{k-m}]$ crosses through the relative old samples, which are assigned descending fuzzy membership grades. As shown in Figure 2, the values of the learning weight at points A and B are larger than that of point C because the states A and B are closer to the current situation. Besides, as

the sampling time window moves, the trapezoidal learning-weight function moves along the time axis accordingly. Therefore, reasonable and adaptable learning weights are assigned for all samples in the whole walking process. The deduced learning weights are then used in the learning algorithm of the TF-LSSVR.

The nonlinear dynamic model $G(\cdot)$ in (21) can be built when the learning process of the TF-LSSVR is completed. After this, considering the parameters of the TF-LSSVR, controller (21) can be written as

$$\tau = G(\Theta, \psi), \quad (24)$$

where τ is the driving torque, Θ is the joint angle, and $\psi = [C, \sigma, m, n, h]^T$ is a parameter vector. Here, we denote the

FIGURE 9: Learning results for Submodel 2 of the SSP (the step length is 0.20 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

plenty coefficient by C , the width of the RBF kernel by σ , and the width of the moving time window by m and n , and we denote the minimum value of the membership function by h . To obtain optimized parameters for the controller, a CKF-based training method is proposed in the next section.

3.3. Training the Parameters of the Controller Using CKF

3.3.1. Parameters Optimization Problem in a Form Suitable for CKF. In order to cast the parameters optimization problem of the TF-LSSVR in a form suitable for CKF, we let the parameters of the TF-LSSVR constitute the state of a nonlinear system, and we let the output of the TF-LSSVR constitute the output of the nonlinear system to which the CKF is applied.

As mentioned above we denote the state of the nonlinear system by

$$\psi = [C, \sigma, m, n, h]^T. \quad (25)$$

The vector ψ thus consists of all of the parameters for the TF-LSSVR arranged in a linear array. Let $\tau = G(\Theta, \psi)$ be the transfer function of the TF-LSSVR, where τ is the output, Θ is the input, and $\psi = [C, \sigma, m, n, h]^T$ is a parameter vector. The training of the TF-LSSVR can be formulated as a filtering problem. The model for updating the parameters using CKF can be written as

$$\begin{aligned} \psi_k &= \psi_{k-1}, \\ \tau_k &= G(\Theta_k, \psi_k). \end{aligned} \quad (26)$$

FIGURE 10: Learning results for Submodel 1 of the DSP (the step length is 0.16 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

Similar to the approach of training fuzzy systems with the extended Kalman filter [26], in order to execute a stable Kalman filter algorithm, we add some process noise and measurement noise to the above model. That is,

$$\begin{aligned}\psi_k &= \psi_{k-1} + w_{k-1}, \\ \tau_k &= G(\Theta_k, \psi_k) + v_k,\end{aligned}\quad (27)$$

where w_{k-1} and v_k are artificially added Gaussian noise sequences with zero means and covariances Q_{k-1} and R_k , respectively.

3.3.2. The CKF Algorithm for Updating the Parameters of the Controller. To update the parameters of the controller, the CKF algorithm [27] is implemented as follows.

Step 1 (cubature-points calculation). The point (η_i, β_i) is called cubature point here. For the third-degree spherical-radial rule, it entails a total of $2u$ cubature points. η_i and β_i can be calculated as follows:

$$\begin{aligned}\eta_i &= \sqrt{u} \langle 1 \rangle_i, \\ \beta_i &= \frac{1}{2u},\end{aligned}\quad (28)$$

where $i = 1, 2, \dots, 2u$ and $\langle 1 \rangle_i$ denotes the i th column of the set $\langle 1 \rangle$. For example, $\langle 1 \rangle = \langle \langle 1, 0 \rangle^T, \langle -1, 0 \rangle^T, \langle 0, -1 \rangle^T, \langle 0, 1 \rangle^T \rangle$ when $u = 2$. u is the dimension of the state vector.

FIGURE 11: Learning results for Submodel 1 of the DSP (the step length is 0.18 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

Step 2 (time update). After calculating the cubature points, we use the cubature-point set to numerically compute the integrals and obtain the time update:

$$\begin{aligned}
 \psi_{i,k-1|k-1} &= \sqrt{P_{k-1|k-1}} \eta_i + \hat{\psi}_{k-1|k-1}, \\
 \psi_{i,k|k-1}^* &= \psi_{i,k-1|k-1} + w_{k-1}, \\
 \hat{\psi}_{k|k-1} &= \sum_{i=1}^{2u} \beta_i \psi_{i,k|k-1}^*, \\
 P_{k|k-1} &= \sum_{i=1}^{2u} \beta_i \psi_{i,k|k-1}^* \psi_{i,k|k-1}^{*T} - \hat{\psi}_{k|k-1} \hat{\psi}_{k|k-1}^T + Q_{k-1}, \\
 \psi_{i,k|k-1} &= \sqrt{P_{k|k-1}} \xi_i + \hat{\psi}_{k|k-1}.
 \end{aligned} \tag{29}$$

Step 3 (measurement update). Consider

$$\begin{aligned}
 \tau_{i,k|k-1} &= G(\Theta_{i,k|k-1}, \psi_{i,k|k-1}) + v_k, \\
 \hat{\tau}_{k|k-1} &= \sum_{i=1}^{2u} \beta_i \tau_{i,k|k-1}, \\
 P_{\tau\tau,k|k-1} &= \sum_{i=1}^{2u} \beta_i \tau_{i,k|k-1} \tau_{i,k|k-1}^T - \hat{\tau}_{k|k-1} \hat{\tau}_{k|k-1}^T + R_k, \\
 P_{\psi\tau,k|k-1} &= \sum_{i=1}^{2u} \beta_i \psi_{i,k|k-1} \tau_{i,k|k-1}^T - \hat{\psi}_{k|k-1} \hat{\tau}_{k|k-1}^T, \\
 W_k &= P_{\psi\tau,k|k-1} P_{\tau\tau,k|k-1}^{-1}, \\
 \hat{\psi}_{k|k} &= \hat{\psi}_{k|k-1} + W_k (\tau_k - \hat{\tau}_{k|k-1}), \\
 P_{k|k} &= P_{k|k-1} - W_k P_{\tau\tau,k|k-1} W_k^T,
 \end{aligned} \tag{30}$$

FIGURE 12: Learning results for Submodel 1 of the DSP (the step length is 0.20 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

where W_k is the Kalman gain, $\hat{\psi}_{k|k}$ is the updated state, and $P_{k|k}$ is the corresponding error covariance. $P_{\psi\tau,k|k-1}$ is the cross-covariance matrix and $P_{\tau\tau,k|k-1}^{-1}$ is the inverse matrix of the innovation covariance matrix.

After time update and measurement update, an estimation of the parameters ψ for the TF-LSSVR is obtained. That is to say, the initial controller with random parameters is updated to the proposed data-based controller.

4. Simulation Research

In this section, we test our proposed learning control method on the control of a seven-link robot by simulation experiments.

Matlab 7.0 is used to model the humanoid robot and the controller.

4.1. Simplified Model of a Seven-Link Humanoid Robot. The simplified model of the robot has two legs and a trunk. Each leg is composed of a thigh, a shank, and a foot. There is one degree of freedom (DOF) in the trunk, one for each hip, one for each knee, and one for each ankle. In this paper, we focus on the sagittal dynamics of the humanoid robot. The simple model of the humanoid is shown in Figure 3. The details of the humanoid can be referenced in Table 1.

4.2. Sampling for the TF-LSSVR Learning. The whole walking cycle is denoted by T_c , and let $T_c = 1$ s with 0.2 s for the

FIGURE 13: Learning results for Submodel 2 of the DSP (the step length is 0.16 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

TABLE 1: Parameters of the humanoid robot.

Link	Length (m)	Mass (kg)	Inertia moment (kg·m ²)
Trunk	0.175	1.060	0.01623
Thigh	0.130	0.075(*2)	0.00037(*2)
Shank	0.130	0.075(*2)	0.00037(*2)
Foot	0.020	0.060(*2)	0.00004(*2)

DSP and 0.8 s for the SSP. The trunk of the humanoid robot is assumed to be upright during walking. Three different step lengths are implemented (0.16 m, 0.18 m, and 0.20 m) and the step height is 0.02 m for all the three step lengths. The sampling interval is 0.025 s. That is to say, there are 40 groups of samples that are collected in a single walking cycle.

To validate the advantage of the proposed method, a simulated time-varying external perturbation $\tau_d = 0.08 \sin(5t)$ (Nm) is considered here for the tests. It is a horizontal external force with duration of 0.1 s that is applied on the hip at $t = 0.15$ s in the DSP and $t = 0.65$ s in the SSP, respectively.

4.3. Parameters and the Learning Results

4.3.1. Experimental Conditions and the Parameters. Here, we use the universal RBF as the kernel for the TF-LSSVR.

The penalty coefficient $C = 1000$ and width of the RBF kernels $\sigma = 0.9$ are determined by a 10-fold cross-validation strategy. The other initial parameters for the proposed framework are identified according to experiences, which include the number of the steps for a full weight $m = 1$, the number of the steps for a gradient weight $n = 1$, and the

TABLE 2: Initial radians of the humanoid joint angles.

Humanoid joint angles (rad)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
Initial radians when step length is 0.16 m	0.0698	0.2418	0.6811	0	0	0.3550	0.5989
Initial radians when step length is 0.18 m	0.0785	0.1819	0.7036	0	0	0.3550	0.5989
Initial radians when step length is 0.20 m	0.0873	0.1183	0.7210	0	0	0.3550	0.5989

FIGURE 14: Learning results for Submodel 2 of the DSP (the step length is 0.18 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

lower bound of the fuzzy membership function $h = 0.5$. Then the initial parameters are updated using CKF.

Parameters for the CKF are as follows: process noise $w_{k-1} \sim (0, 10^{-3})$ and the measurement noise $v_k \sim (0, 10^{-3})$, where w_{k-1} and v_k are both white, zero-mean, uncorrelated noises. The initial covariance of the humanoid state is $P_0 = \text{diag}\{10^{-3}, \dots, 10^{-3}\}_{7 \times 7}$. The initial radians of the humanoid joint angles are shown in Table 2.

4.3.2. Learning Results of the TF-LSSVR. The learning results of the proposed TF-LSSVR are compared with two other

intelligent methods including fuzzy and the traditional LS-SVR. The designing details of the fuzzy control system can be found in the literature [14]. Nonlinear dynamics of the submodels formulated in (19)-(20) are learned using the three different methods (see Figures 4–15).

Define control error $e = \hat{\tau} - \tau$, and the integral square error (ISE) is defined as the measure index:

$$\text{ISE} = \frac{1}{N} \sum_{i=1}^N (\hat{\tau}^{(i)} - \tau^{(i)})^2, \quad (31)$$

FIGURE 15: Learning results for Submodel 2 of the DSP (the step length is 0.20 m). (a) Fuzzy; (b) LS-SVR; (c) the proposed TF-LSSVR.

TABLE 3: Comparisons of the ISE performances.

	$e_{\text{sup.hip}}$	$e_{\text{sup.ankle}}$	$e_{\text{left.knee}}$	$e_{\text{right.knee}}$
Fuzzy	$8.9e-002$	$9.6e-002$	$9.2e-001$	$9.9e-002$
LS-SVR	$7.9e-002$	$5.4e-002$	$2.8e-001$	$7.3e-002$
TF-LSSVR	$6.1e-006$	$7.2e-006$	$1.0e-004$	$2.1e-005$

where i denotes the sampling index and $N = 40$ is the number of all the samples. $\hat{\tau}^{(i)}$ is the output of the learning methods and $\tau^{(i)}$ is the desired output. The integrated square errors (ISEs) criterion is listed in Table 3, which indicates that the proposed TF-LSSVR achieves a better learning result than the other two existing methods.

4.4. Performance Comparisons. ZMP trajectory with disturbance in the SSP can be found in Figures 16–18, which

indicate that the proposed learning control method produces a larger stability margin relative to the traditional ones. Similar situations appeared in the DSP and the comparisons of humanoid walking when disturbances occur in the DSP are shown in Figures 19–21. As we can see, the ZMP trajectories corresponding to the proposed method have bigger stability margins for the humanoid compared to the other two intelligent methods. That is to say, using the data online and offline with different weights, the proposed learning control method is more effective in learning the external disturbances and increasing the stability margin of humanoid robots.

5. Conclusions

In this work, a TF-LSSVR-based control system is proposed to learn the external disturbances and increase the stability

FIGURE 16: ZMP trajectory with disturbance in the SSP when the step length is 0.16 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

FIGURE 19: ZMP trajectory with disturbance in the DSP when the step length is 0.16 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

FIGURE 17: ZMP trajectory with disturbance in the SSP when the step length is 0.18 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

FIGURE 20: ZMP trajectory with disturbance in the DSP when the step length is 0.18 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

FIGURE 18: ZMP trajectory with disturbance in the SSP when the step length is 0.20 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

FIGURE 21: ZMP trajectory with disturbance in the DSP when the step length is 0.20 m. Fuzzy controller (green); LS-SVR controller (magenta); the proposed TF-LSSVR controller (black).

margin of humanoid robots. First, data for training the controller is collected by implementing simulation experiments. Secondly, a TF-LSSVR with a time-related TFMF is proposed to train the controller using the simulated data. Thirdly, the parameters of the proposed TF-LSSVR are updated using a CKF. Simulation results are provided.

The proposed method is shown to be effective in learning and adapting occasional external disturbances and ensuring the stability margin of the robot. We believe that the proposed method will be very promising for the development of autonomous humanoid robots.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Project 61403264 and Project 61305098 and the Natural Science Foundation of Guangdong Province under Project 2016A030310018.

References

- [1] J.-W. Heo and J.-H. Oh, "Biped walking pattern generation using an analytic method for a unit step with a stationary time interval between steps," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1091–1100, 2015.
- [2] T.-H. S. Li, P.-H. Kuo, Y.-F. Ho, M.-C. Kao, and L.-H. Tai, "A biped gait learning algorithm for humanoid robots based on environmental impact assessed artificial bee colony," *IEEE Access*, vol. 3, pp. 13–26, 2015.
- [3] P. Hénaff, V. Scesa, F. Ben Oueddou, and O. Bruneau, "Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance: experiments on the standing posture," *Control Engineering Practice*, vol. 19, no. 1, pp. 89–99, 2011.
- [4] C. Sabourin and O. Bruneau, "Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks," *Robotics and Autonomous Systems*, vol. 51, no. 2-3, pp. 81–99, 2005.
- [5] H.-K. Shin and B. K. Kim, "Energy-efficient gait planning and control for biped robots utilizing vertical body motion and allowable zmp region," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2277–2286, 2015.
- [6] M. Rakovic, B. Borovac, M. Nikolic, and S. Savic, "Realization of biped walking in unstructured environment using motion primitives," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1318–1332, 2014.
- [7] J. P. Ferreira, M. M. Crisostomo, and A. P. Coimbra, "Adaptive PD controller modeled via support vector regression for a biped robot," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 941–949, 2013.
- [8] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs," *IEEE Access*, vol. 3, no. 1, pp. 323–332, 2015.
- [9] M. Vukobratovic and B. Borovac, "Zero-moment point—thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.
- [10] M. Vukobratović, B. Borovac, and V. Potkonjak, "ZMP: a review of some basic misunderstandings," *International Journal of Humanoid Robotics*, vol. 3, no. 2, pp. 153–175, 2006.
- [11] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1321–1326, Leuven, Belgium, May 1998.
- [12] T.-H. S. Li, Y.-T. Su, S.-H. Liu, J.-J. Hu, and C.-C. Chen, "Dynamic balance control for biped robot walking using sensor fusion, Kalman filter, and fuzzy logic," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4394–4408, 2012.
- [13] H.-K. Shin and B. K. Kim, "Energy-efficient gait planning and control for biped robots utilizing the allowable ZMP region," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 986–993, 2014.
- [14] Z. Liu, Y. Zhang, and Y. Wang, "A type-2 fuzzy switching control system for biped robots," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1202–1213, 2007.
- [15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [16] H. Drucker, C. J. C. Surges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proceedings of the 10th Annual Conference on Neural Information Processing Systems (NIPS '96)*, pp. 155–161, December 1996.
- [17] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1–4, pp. 85–105, 2002.
- [18] S. Chatterjee, "Vision-based rock-type classification of limestone using multi-class support vector machine," *Applied Intelligence*, vol. 39, no. 1, pp. 14–27, 2013.
- [19] L. Guo, Y. Wu, L. Zhao, T. Cao, W. Yan, and X. Shen, "Classification of mental task from EEG signals using immune feature weighted support vector machines," *IEEE Transactions on Magnetics*, vol. 47, no. 5, pp. 866–869, 2011.
- [20] J. P. Ferreira, M. M. Crisostomo, and A. P. Coimbra, "SVR versus neural-fuzzy network controllers for the sagittal balance of a biped robot," *IEEE Transactions on Neural Networks*, vol. 20, no. 12, pp. 1885–1897, 2009.
- [21] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 4, pp. 438–447, 2010.
- [22] L. Wang, Z. Liu, C. L. P. Chen, Y. Zhang, S. Lee, and X. Chen, "Energy-efficient SVM learning control system for biped walking robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 831–837, 2013.
- [23] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended Kalman algorithm," in *Advances in Neural Information Processing Systems I. Denver 1988*, D. S. Touretzky, Ed., pp. 133–140, Morgan Kaufmann, San Mateo, Calif, USA, 1989.
- [24] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of non-linear dynamical systems with Kalman filter trained recurrent networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 279–297, 1994.
- [25] T. Jiang and Y. Li, "Generalized defuzzification strategies and their parameter learning procedures," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, pp. 64–71, 1996.
- [26] D. Simon, "Training fuzzy systems with the extended Kalman filter," *Fuzzy Sets and Systems*, vol. 132, no. 2, pp. 189–199, 2002.

- [27] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [28] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: theory and simulations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, 2010.
- [29] W. Li and Y. Jia, "Location of mobile station with maneuvers using an IMM-based cubature Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4338–4348, 2012.
- [30] I. Arasaratnam and S. Haykin, "Cubature Kalman smoothers," *Automatica*, vol. 47, no. 10, pp. 2245–2250, 2011.
- [31] S. Haykin, A. Zia, Y. Xue, and I. Arasaratnam, "Control theoretic approach to tracking radar: first step towards cognition," *Digital Signal Processing: A Review Journal*, vol. 21, no. 5, pp. 576–585, 2011.
- [32] M. Vukobratović, B. Borovac, D. Surla, and D. Stokić, *Humanoid Locomotion: Dynamic, Stability, Control and Application*, Springer, Berlin, Germany, 1990.

Hindawi

Submit your manuscripts at
<http://www.hindawi.com>