Finite-time Tracking Control with Velocity Constraints for the Stochastic Rehabilitative Training Walker System Considering Different Rehabilitee Masses

Ping Sun (✉ tonglongsun@sut.edu.cn)  
Shenyang University of Technology  https://orcid.org/0000-0002-4131-6648

Shuoyu Wang  
Kochi University of Technology

Rui Shan  
Shenyang University of Technology

Research Article

Keywords: random rehabilitee mass variation, velocity constraint, rehabilitative training walker, stochastic finite-time tracking control

Posted Date: August 5th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-278158/v1

License: 😊 This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License

Version of Record: A version of this preprint was published at Nonlinear Dynamics on November 12th, 2021. See the published version at https://doi.org/10.1007/s11071-021-06912-3.
Finite-time Tracking Control with Velocity Constraints for the Stochastic Rehabilitative Training Walker Systems Considering Different Rehabilitee Masses

Ping Sun · Shuoyu Wang · Rui Shan

Abstract This study discusses a finite-time tracking controller for a rehabilitative training walker that imposes velocity constraints. The walker was described using a stochastic model through which the rehabilitee mass can randomly change, and a velocity constraint method was proposed to control the velocity input to every omniwheel based on a model predictive algorithm. This approach is novel in that the velocity constraint information obtained from the kinematics model was used to design the tracking controller based on the stochastic dynamic model, thus successfully constraining the actual velocity of walker as per the stochastic system. The nonlinear tracking controller was built for the stochastic rehabilitative walker to make the system’s finite time stable. Also, simulation and experiment were performed, and results confirmed that the proposed tracking control method with velocity constraints is very effective, so it may enable various rehabilitees to train safely.

Keywords random rehabilitee mass variation · velocity constraint · rehabilitative training walker · stochastic finite-time tracking control

1 Introduction

In recent years, the use of rehabilitative training robots in helping disabled people with walking difficulties has significantly increased. For example,
kle rehabilitation robots are characterized with their ability to provide long-term accurate and uniform training [1], and they are mainly classified into the following two categories based on the mobility of the used device during operation: wearable robots and platform-based ones [2]. By potentially initiating neuroplastic changes, robot-assisted gait training has shown to be a promising neurorehabilitation intervention method with regard to improving gait recovery for patients who have been through strokes or brain injuries [3]. Lower limb rehabilitation robots with remote control systems are designed to assist patients in walking, where the remote system is based on a client/server network framework to realize the remote control of the robot state logic and the transmission of patient training data [4]. Also, a lower limb exoskeleton, which is a major class of rehabilitation robots, is a wearable device that is based on bionics, mechanical engineering, and artificial intelligence disciplines [5]. In addition, it can connect with the human bodies in a wearable manner and can control the movement of all joints in the training process [6].

The above-mentioned robots can typically be programmed to follow specific training trajectories for a wide range of injuries. Trajectory tracking control is thus a topic of active study, and various schemes have been proposed based on it to enhance tracking performance, such as robust nonlinear control for robotic exoskeletons [7], multiple-input and multiple-output sliding mode control for gait exoskeletons [8], and redundant input safety tracking control for training walkers [9]. However, only a few of these studies have focused on utilizing randomly changeable system parameters.

In practice, robot systems are often subject to unpredictable effects due to random variations in their working environments, which can substantially reduce their tracking precision [10-12]. An adaptive tracking control scheme has been proposed for the flexible-joint robots experiencing random noise [13], and some researchers have also considered the tracking control of stochastic Lagrangian manipulator systems [14, 15] and stochastic optimal control schemes for the underwater robots involving Gaussian processes [16]. However, these studies have mainly focused on the random disturbances from external environments and ignored the random variations in internal system parameters.

In reality, robot systems often involve many changing parameters. Thus, an iterative learning control method with robust conditions has been developed to deal with load variations [17]. Also, a stochastic tracking scheme for rehabilitative walkers that consider random changes with regard to interaction forces and center of gravity has been studied [18]. Although this work studied randomly varying system parameters, it neglected the changes in a few factors, such as rehabilitee mass and goods load. Generally, the dynamics models used to design robot control systems [19-21] place the rehabilitee mass or the loads grabbed by manipulators in the coefficient matrix before acceleration. This means that the mass/load cannot be regarded as an external disturbance with a bounded range because the robot does not account for the fact that it may change [22, 23]. Hence, investigating methods for handling random mass/load variations is still challenging.
Rehabilitation robots are also operated in complicated environments, so their velocities inevitably change. Unlike many other mechanical systems, rehabilitation robots must guarantee that all human-robot interactions are safe. As is well-known, when a robot system is described using a stochastic model, the system states are given in terms of expected values. Unfortunately, this means that only the robot’s mean velocity can be constrained and not its instantaneous velocity. Thus, traditional approaches, such as adding a saturation function [24], using a barrier Lyapunov function [25], or applying an active velocity constraint algorithm [26], are unsuitable for dealing with the real-time velocity constraints of stochastic systems. In the case of random variations with regard to the mass of a rehabilitee, it is nontrivial to design a stochastic finite-time controller that can prevent exceeding safety boundaries in the stage of transient motion and achieve fast precise tracking with velocity constraints.

Currently, unlike conventional system state constraint strategies, model predictive control (MPC) methods are proving to be effective in addressing constraint problems, and they have been applied in multiple fields [27-30]. Even so, we do not yet know how to use MPC to restrict a robot’s velocity, incorporate velocity constraints into the stochastic dynamic model of a rehabilitative training walker, or achieve stochastic finite-time tracking control under velocity constraints. These are clearly fundamental issues in the robot control theory, so they are worth exploring more deeply. To the best of our knowledge, no works have been performed on velocity constraints and finite-time control for stochastic robot systems, which motivated this study.

The above-mentioned issues indicate that there are several problems worth of future theoretical studies and consideration in engineering applications. Here, we focused on an omnidirectional rehabilitative training walker (ODW) [31], which is a mobile robot. The rehabilitation doctor makes a training plan according to the walking ability of the trainer, and the rehabilitees improve their walking ability and balance ability through walking training. Compared with other related studies, in this study, several novel contributions were made as summarized below.

(i) Differing rehabilitee masses present major challenges for human-robot systems. Here, we established a new stochastic model, where we modeled different rehabilitee masses by a random parameter rather than by simply considering a fixed value.

(ii) The walker’s velocity is carefully restricted using the ODW’s kinematics model. We propose to constrain the velocity using a model predictive method that controls the velocity input to every omniwheel. This helped in revealing the relationship between the kinematics model that constrains the velocity and the dynamics model that controls trajectory tracking.

(iii) We developed a tracking controller that accounts for differing rehabilitee masses and constrains velocity such that the stochastic trajectory tracking error system is finite-time stable. We also discussed the proposed algorithm’s effectiveness for the ODW.
The rest of this paper is organized as follows. In Section 2, we formulate a stochastic ODW model that allows for random rehabilitee mass variations, and in Section 3, we explain the velocity constraints algorithm. In Section 4, we discuss the stochastic tracking controller and analyze its finite-time stability. Finally, we present the conducted simulation and experimental results in Sections 5 and 6, respectively, and the remarks are in Section 7.

2 Dynamic ODW model with randomly varying masses

Fig. 1 [32] illustrates the ODW design, and Fig. 2 [32] shows the coordinate system and considered parameters.

\[ M_0 \dot{\mathbf{X}}(t) + M_0 \ddot{\mathbf{X}}(t) = B(\theta)u(t) \] (1)

where

\[
M_0 = \begin{bmatrix}
M + m & 0 & 0 \\
0 & M + m & 0 \\
0 & 0 & I_0 + m r_0^2
\end{bmatrix}, \quad K = \begin{bmatrix}
1 & 0 & p \\
0 & 1 & q \\
0 & 0 & 1
\end{bmatrix}, \quad X(t) = \begin{bmatrix}
x(t) \\
y(t) \\
\theta(t)
\end{bmatrix}
\]

\[
B(\theta) = \begin{bmatrix}
-\sin \theta_1 & \sin \theta_2 & \sin \theta_3 & -\sin \theta_4 \\
\cos \theta_1 & -\cos \theta_2 & \cos \theta_3 & \cos \theta_4 \\
\lambda_1 & -\lambda_2 & -\lambda_3 & \lambda_4
\end{bmatrix}, \quad \begin{aligned}
\lambda_1 &= l_1 \cos(\theta_1 - \phi_1) \\
\lambda_2 &= l_2 \cos(\theta_2 - \phi_2) \\
\lambda_3 &= l_3 \cos(\theta_3 - \phi_3) \\
\lambda_4 &= l_4 \cos(\theta_4 - \phi_4)
\end{aligned}
\]
Here, $M$ is the ODW’s mass, $m$ is the user’s mass, and $I_0$ is the ODW’s inertial mass. In addition, $f_1, f_2, f_3,$ and $f_4$ are the input forces, and $\theta$ is the angle between $x'$ axis and the first omniwheel, denoted by $\theta = \theta_1$. Likewise, we have $\theta_2 = \theta + \frac{\pi}{2}, \theta_3 = \theta + \pi$, and $\theta_4 = \theta + \frac{3\pi}{2}$ for the other three omniwheels.

To construct a stochastic model that accounts for the differing masses of different rehabilitees, we replaced the parameter $m$ in equation (1) by $m_s : m = m_s + \Delta m$, where $m_s$ is constant, and $\Delta m$ represents the changes in $m$ between the rehabilitees. The revised model is then

$$
\dot{X}(t) = M_s^{-1}B(\theta)u(t) + M_s^{-1}B_s(\theta)\varsigma(t) \tag{2}
$$

where

$$
M_s = \begin{bmatrix}
M + m_s & 0 & 0 \\
0 & M + m_s & 0 \\
0 & 0 & I_0 + m_s r_0^2
\end{bmatrix},
$$

$$
B_s(\theta) = \frac{1}{2}(M + m) \begin{bmatrix}
-\sin \theta - \cos \theta - \cos \theta & \sin \theta & \frac{2}{M + m} \\
\cos \theta - \sin \theta & 0 & 0 \\
0 & 0 & 0
\end{bmatrix},
$$

$$
\varsigma(t) = \begin{bmatrix}
\dot{\theta}(t)(\lambda_1 - \lambda_3) \\
\dot{\theta}(t)(\lambda_2 - \lambda_4) \\
\ddot{\theta}(t)(\lambda_1 - \lambda_3) \\
\Delta m \ddot{x}(t) \\
\Delta m \ddot{y}(t) \\
\Delta m r_0^2 \ddot{\theta}(t)
\end{bmatrix}.
$$

Here, the random parameter $\varsigma(t)_{\gamma = 1}$ has the properties of white noise [33]. Following [34] (Section 5.5), we replaced $\varsigma(t)$ by $\frac{d\theta}{dt}$, enabling us to achieve the stochastic differential equation (SDE) of model (2) as follows:

$$
d\dot{X}(t) = M_s^{-1}B(\theta)u(t)dt + M_s^{-1}B_s(\theta) \circ d\theta \tag{3}
$$

where $\theta$ is an independent Wiener process.

If we let $M_s^{-1}B_s(\theta) = [\eta_{\rho d}]_{3 \times 7}$ ($\rho = 1, 2, 3; d = 1, 2, \cdots, 7$), the Wong-Zakai correction term is

$$
\frac{1}{2} \begin{bmatrix}
\sum_{\delta = 1}^{7} \left( \frac{\partial \eta_{1 \delta}}{\partial x(t)} \cdot \eta_{1 \delta} + \frac{\partial \eta_{1 \delta}}{\partial y(t)} \cdot \eta_{2 \delta} + \frac{\partial \eta_{1 \delta}}{\partial \theta(t)} \cdot \eta_{3 \delta} \right) \\
\sum_{\delta = 1}^{7} \left( \frac{\partial \eta_{2 \delta}}{\partial x(t)} \cdot \eta_{1 \delta} + \frac{\partial \eta_{2 \delta}}{\partial y(t)} \cdot \eta_{2 \delta} + \frac{\partial \eta_{2 \delta}}{\partial \theta(t)} \cdot \eta_{3 \delta} \right) \\
\sum_{\delta = 1}^{7} \left( \frac{\partial \eta_{3 \delta}}{\partial x(t)} \cdot \eta_{1 \delta} + \frac{\partial \eta_{3 \delta}}{\partial y(t)} \cdot \eta_{2 \delta} + \frac{\partial \eta_{3 \delta}}{\partial \theta(t)} \cdot \eta_{3 \delta} \right)
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} \tag{4}
$$
The corresponding Itô SDE of model (3) is given by

\[ d\dot{X}(t) = M_s^{-1}B(\theta)u(t)dt + M_s^{-1}B_s(\theta)d\vartheta \]  \hspace{1cm} (5)

We assumed that the white noise \( \zeta(t) \) has a power spectral density of \( \frac{H}{2\pi} \), i.e., \( d\vartheta = Hd\varphi \). Thus, model (38) becomes

\[ d\dot{X}(t) = M_s^{-1}B(\theta)u(t)dt + M_s^{-1}B_s(\theta)\Pi d\varphi \]  \hspace{1cm} (6)

By defining \( \dot{X}(t)^{3\times1} = v(t)^{3\times1} = [v_x(t) \ v_y(t) \ v_\theta(t)]^T \) as the actual velocities of x-axis, y-axis, and the orientation angle, respectively, model (6) can be expressed as follows:

\[
\begin{cases}
    dX(t) = v(t)dt \\
    dv(t) = M_s^{-1}B(\theta)u(t)dt + M_s^{-1}B_s(\theta)\Pi d\varphi
\end{cases}
\]  \hspace{1cm} (7)

**Remark 1:** In general, the mass \( m \) (e.g., the rehabilitee mass or the load grabbed by a manipulator) is included in the system model and was regarded as being fixed when designing the controller. However, in practice, it can vary between different rehabilitees or items. Thus, using such fixed values in practical implementations can cause issues and is unlikely to achieve optimal system performance. Here, we instead decomposed the mass \( m \) as \( m = m_s + \Delta m \) with a constant \( m_s \) and a random variable \( \Delta m \), resulting in a stochastic model with a randomly varying mass, as shown in equation (7).

**Remark 2:** Rehabilitees form a part of a closed rehabilitative robot system, meaning that their differing masses have a crucial impact on a system’s tracking performance. It is unrealistic to adjust the controller parameters for each rehabilitee to achieve precise tracking. Instead, it is more reasonable to design a controller that can train arbitrary rehabilitees. In this study, we explored a new approach based on a stochastic dynamic model (7), that can capture the random mass variations between rehabilitees.

**Remark 3:** Our stochastic model (7), which accounts for random mass variations, is more general than the one presented in [32] (their model (11)). When \( \Delta m = 0 \), \( M_s \) is reduced to its coefficient matrix \( M_0 \), making its model a special case of the proposed model.

Next, we made the following reasonable assumption.

**Assumption 1:** According to the meaning of the system’s physical quantities, a constant \( h \) exists, such that

\[ \|M_s^{-1}\|_F^2 \|(M + m)^2 + 3\|\Pi\|_F^2 \leq h \]  \hspace{1cm} (8)

**Remark 4:** Assumption (8) is reasonable because the ODW’s mass \( M \) and the user’s mass \( m \) in matrix \( B_s(\theta) \) are bounded, and the matrix’s Frobenius norm is induced by the Euclidean norm, meaning that the norms are compatible.
3 Model predictive velocity constraints algorithm

It would be very difficult to directly limit the ODW’s velocity in practice because we allow for random rehabilitee mass variations. Generally, random variations are described in terms of their expectation values. In other words, the ODW’s mean velocity can be restricted using a suitable method, but its instantaneous movement speed cannot be directly limited using the stochastic model (7).

In this study, the ODW’s velocity $\dot{X}(t)$ is constrained using the following kinematics model [35]:

$$V(t) = K_G(t)\dot{X}(t)$$  \hspace{1cm} (9)

Here,

$$K_G(t) = \begin{bmatrix} -\sin \theta_1(t) & \cos \theta_1(t) & \lambda_1 \\ \sin \theta_2(t) & -\cos \theta_2(t) & -\lambda_2 \\ \sin \theta_3(t) & -\cos \theta_3(t) & -\lambda_3 \\ -\sin \theta_4(t) & \cos \theta_4(t) & \lambda_4 \end{bmatrix}$$

and $V(t)_{4 \times 1} = [v_1(t) \ v_2(t) \ v_3(t) \ v_4(t)]^T$, where $v_i(t)$ denotes the velocity inputs to each omniwheel.

Then, equation (9) can be expressed as follows:

$$\dot{X}(t) = K_P^{-1}(t)V(t)$$  \hspace{1cm} (10)

where $K_P^{-1}(t)_{3 \times 4} = (K_G^T \cdot K_G)^{-1} \cdot K_G^T$ is the pseudoinverse matrix of $K_G(t)$.

By using a nonlinear feed-forward difference method and rewriting the velocity input $V(t)$ in the incremental form, the discretized model (10) can be written as follows given the system state $X(t)$ and by expressing the output as $Y(t) = X(t)$:

$$\begin{cases} X(k + 1) = AX(k) + BV(k) \\ V(k) = V(k - 1) + \Delta V(k) \\ Y(k) = \dot{X}(k) = BV(k) \end{cases}$$  \hspace{1cm} (11)

where $k = 0, 1, \cdots, N - 1$, $N$ is the prediction time horizon, $\Delta V(k)_{4 \times 1}$ is the current velocity input increment, and $V(k - 1)_{4 \times 1}$ is the velocity input at the previous timestep. Also, $A_{3 \times 3} = I_3$ and $B_{3 \times 4} = T \cdot K_P^{-1}(t)$, where $I_3$ denotes the identity matrix and $T$ is the sampling period.

Next, the following constraints on the predicted velocity $\dot{X}$ in the $x$-axis, $y$-axis, and orientation angle, respectively, in addition to the velocity input $V$ to each omniwheel can be obtained as follows:

$$\begin{cases} \dot{X}_{\min} \leq \dot{X} \leq \dot{X}_{\max} \\ \dot{V}_{\min} \leq \dot{V} \leq \dot{V}_{\max} \end{cases}$$  \hspace{1cm} (12)

where $\dot{V}_{4N_C \times 1} = [V^T(k \mid k) \ V^T(k + 1 \mid k) \ \cdots \ V^T(k + N_C - 1 \mid k)]^T$ represents the predicted velocity inputs at the future timesteps from $k$ to $k + N_C - 1$, $N_C$ is the time horizon for control, $\dot{X}_{3N \times 1} = [\dot{X}^T(k \mid k) \ \dot{X}^T(k + 1 \mid k) \ \cdots \ \dot{X}^T(k + N - 1 \mid k)]^T$ represents the predicted actual velocities at the future timesteps from $k$ to
\( k + N - 1 \), \( \dot{V}_{\text{max}} \) is the upper velocity input bound, and \( \dot{V}_{\text{min}} = -\dot{V}_{\text{max}} \). In addition, \( \dot{X}_{\text{max}} \) is the upper velocity bound, and \( \dot{X}_{\text{min}} = -\dot{X}_{\text{max}} \).

Then, based on model (11), the following velocity prediction model can be obtained:

\[
\dot{\bar{X}} = \Phi V(k - 1) + G\Delta \bar{V} \quad (13)
\]

where \( \Delta V_{\bar{X}} = \begin{bmatrix} \Delta V^T(k|k) & \Delta V^T(k+1|k) & \cdots & \Delta V^T(k+N-1|k) \end{bmatrix}^T \), \( \Phi_{3N \times 4} = B_pL_0 \), \( G_{3N \times 4N} = B_pL_1 \),

\[
B_p = \begin{bmatrix} B(k) & 0 & \cdots & 0 \\
0 & B(k+1) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & B(k+N-1) \end{bmatrix}_{3N \times 4N}
\]

\[
L_0 = \begin{bmatrix} I_4 \\
I_4 \\
\vdots \\
I_4 \end{bmatrix}_{4N \times 4} \quad , \quad L_1 = \begin{bmatrix} I_4 & 0 & \cdots & 0 \\
I_4 & I_4 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
I_4 & I_4 & \cdots & I_4 \end{bmatrix}_{4N \times 4N}
\]

Here, \( B(k+i)_{3 \times 4}, i = 0, 1 \cdots N - 1 \) can be obtained from matrix \( B \) in equation (11), and \( I_4 \) denotes the identity matrix.

By substituting equation (13) into equation (12), the constraints can be expressed in terms of the change in the velocity input \( \Delta \bar{V} \) as follows:

\[
\begin{align*}
& b_{1\min} \leq G\Delta \bar{V} \leq b_{1\max} \\
& b_{2\min} \leq L_1\Delta \bar{V} \leq b_{2\max}
\end{align*} \quad (14)
\]

where

\[
\begin{align*}
b_{1\min} &= \dot{X}_{\text{min}} - \Phi V(k - 1) \\
b_{1\max} &= \dot{X}_{\text{max}} - \Phi V(k - 1) \\
b_{2\min} &= \dot{V}_{\text{min}} - L_0V(k - 1) \\
b_{2\max} &= \dot{V}_{\text{max}} - L_0V(k - 1)
\end{align*}
\]

Also,

\[
G_0\Delta \bar{V} \leq b \quad (15)
\]

where \( G_0_{14N \times 4N} = [-G^T \quad G^T - L_1^T \quad L_1^T]^T \) and \( b_{14N \times 1} = [b_{1\min}^T \quad b_{1\max}^T - b_{2\min}^T \quad b_{2\max}^T]^T \).

Next, a system performance index that minimizes the quadratic velocity tracking error over the prediction time horizon and the velocity input increment over the control time horizon needed to be selected. Hence, we defined the objective function as follows:

\[
J = \min[(\dot{X} - \dot{X}_d)^TQ(\dot{X} - \dot{X}_d) + \Delta \bar{V}^TR\Delta \bar{V}] \quad (16)
\]
where \( \hat{X}_{d(3N \times 1)} = \left[ X_d^T(k | k) \ X_d^T(k + 1 | k) \ \cdots \ X_d^T(k + N - 1 | k) \right]^T \) represents the designated motion velocities, and \( Q \) and \( R \) are diagonal positive definite matrices that are used to tune the performance index \( J \). By substituting equation (13) into equation (16), the objective function can be rewritten as follows:

\[
J = \min \left( \frac{1}{2} \Delta \hat{V}^T H \Delta \hat{V} + c^T \Delta \hat{V} \right)
\]

(17)

where \( H_{4N \times 4N} = 2(G^T Q G + R) \) and \( c_{4N \times 1} = 2G^T Q (\Phi V(k - 1) - \dot{X}_d) \).

Thus, we converted solving an optimization problem with constraints on the velocity \( \dot{X} \) into solving the quadratic programming problem described by equations (15) and (17). Now, the velocity constraints can be implemented as follows.

**Step 0:** Set \( k = 0 \) and initialize \( X(0), V(-1), \dot{X}_{\text{max}}, \text{and} \ V_{\text{max}} \).

**Step 1:** For the current timestep \( k \), calculate \( G_0 \) and \( b \). Solve the optimization problem in equations (15) and (17) to obtain a sequence of the velocity input increments \( \Delta \hat{V} \).

**Step 2:** Apply the first velocity input increment \( \Delta \hat{V}(1 | k) \) in \( \Delta \hat{V} \) to model (11) to obtain the velocity input \( V(k) \) to each omniwheel. Then, apply this to the ODW kinematics model (10) to obtain the actual constrained velocities \( \dot{X}(t) \) in \( x \)-axis, \( y \)-axis, and the orientation angle, respectively.

**Step 3:** Set \( k = k + 1 \) and return to Step 1.

**Remark 5:** To improve the ODW’s tracking performance, the different rehabilitee masses should be considered when designing a control system. However, given the random variation \( \Delta m \), it is almost impossible to constrain the actual velocity \( \dot{X}(t) \) using the stochastic model (7) because random variations are described in terms of expectation values. Thus, only the mean velocity can be constrained and not the instantaneous velocity. Therefore, we instead handled the velocity constraints using the model predictive method, which is a promising approach to solving state constraint problems for stochastic systems.

**Remark 6:** In this study, we used the kinematics model and proposed a model predictive method that limits the ODW’s velocity by controlling the velocity input to each omniwheel. This has the advantage of not only successfully implementing velocity constraints but also providing a way to apply both the kinematics and dynamics models.

### 4 Stochastic finite-time tracking controller

To begin with the main results, the following Lemmas were presented.

**Lemma 1** [36]: Considering the nonlinear system \( \dot{x} = f(x), f(0) = 0, \ x \in R^n \) and by supposing that there is a \( C^1 \) function \( V(x) \) defined in the neighborhood \( \hat{U} \subset R^n \) of the origin and that there are real positive numbers \( d, d_c, \) and \( 0 < \delta < 1 \), then

(i) \( V(x) \) is positive definite in \( \hat{U} \):

(ii) \( \dot{V}(x) \leq -dV^2(x) + d_c, \forall x \in \hat{U} \).

Thus, the origin of the nonlinear system is locally practical finite-time stable.
was introduced with \( n \) into equations (20) and (21), the tracking error system was rewritten as follows:

\[
\sum_{i=1}^{n} k_i |x_i|^{p_i} \leq V(x, t) \leq \sum_{i=1}^{n} k_i' |x_i|^{p_i}
\]

\[
LV(x(t), t) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x}f(x(t), t) + \frac{1}{2} \text{Tr} \left\{ g^T(x(t), t) \frac{\partial^2 V}{\partial x^2} g(x(t), t) \right\}
\]

\[
LV(x, t) = -dV(x, t) + d_c
\]

Then, a unique strong solution \( x(t) = x(t; x_0; t_0) \) to the stochastic nonlinear system exists for each \( x(t_0) = x_0 \in \mathbb{R}^n \), and the system is \( p \)-th-moment exponentially practically stable, where \( p=\min\{p_1, \ldots, p_n\} \), \( \frac{\partial V}{\partial x}, \frac{\partial V}{\partial x^2}, \ldots, \frac{\partial V}{\partial x^n} \in \mathbb{R}^{n \times n} \), and \( \frac{\partial^2 V}{\partial x^2} = \left( \frac{\partial^2 V}{\partial x_i \partial x_j} \right)_{n \times n} \).

By Supposing that the desired and actual motion trajectories are \( X_d(t) \) and \( X(t) \), respectively, the trajectory and velocity tracking errors can be defined as follows:

\[
e_1(t) = X(t) - X_d(t) \quad (18)
\]

\[
e_2(t) = \dot{X}(t) - \dot{X}_d(t) - \alpha e_1(t) = v(t) - \dot{X}_d(t) - \alpha e_1(t) \quad (19)
\]

where \( \dot{X}(t)_{3 \times 1} = v(t)_{3 \times 1} \) is the constrained velocity designed in Section 3, the parameter \( \alpha \) is defined below, and the following vector is defined: \( \text{Sig}(\Sigma)^{n \times 1} = [\Sigma_1^\nu sgn(\Sigma_1), \ldots, \Sigma_n^\nu sgn(\Sigma_n)]^T \), \( \Sigma_{n \times 1} = [\Sigma_1, \ldots, \Sigma_n]^T \in \mathbb{R}^n \), \( 0 < \nu < 1 \).

The goal was to design a finite-time tracking controller that enforces the velocity constraints and handles random rehabilitate mass variations while ensuring that the tracking errors \( e_1(t) \) and \( e_2(t) \) are sufficiently small. By combining equations (18), (19), and (7), we obtained the following system for the tracking errors:

\[
d_e1(t) = [e_2(t) + \alpha e_1(t)]dt \quad (20)
\]

\[
d_e2(t) = [M_s^{-1}B(\theta)u(t) - \dot{X}_d(t) - \alpha e_2(t) - \alpha^2 e_1(t)]dt + M_s^{-1}B_s(\theta)\Pi d\phi \quad (21)
\]

An auxiliary variable \( \chi(e_1(t))_{3 \times 1} \) was introduced with \( \chi(0) = 0 \), and an error variable was defined as follows: \( z(t)_{3 \times 1} = e_2(t) - \chi(e_1(t)) \). By substituting \( z(t)_{3 \times 1} \) into equations (20) and (21), the tracking error system was rewritten as follows:

\[
\begin{cases}
d_e1(t) = [z(t) + \chi(e_1(t))] + \alpha e_1(t)]dt \\
dz(t) = [M_s^{-1}B(\theta)u(t) - \dot{X}_d(t) - \alpha e_2(t) + \alpha \chi(e_1(t))] \\
-\alpha^2 e_1(t) - \chi(e_1(t))]dt + M_s^{-1}B_s(\theta)\Pi d\phi
\end{cases} \quad (22)
\]

As shown in the following section, a design tracking controller was designed for the stochastic ODW system (13) to realize finite-time stability using a backstepping procedure.
**Theorem 1**: For an ODW that can handle arbitrary rehabilitee masses, as expressed by model (7), the stochastic tracking controller with velocity constraints was designed as follows:

\[
u(t) = -\frac{\dot{B}^{-1}(\theta)M_x[\chi(\tilde{x}(t)) + \dot{X}_d(t)] + \alpha \tilde{x}(t) + \alpha^2 c_1(t) + \alpha c_1(t)}{2\frac{\dot{B}^{-1}(\theta)M_x[\chi(\tilde{x}(t)) + \dot{X}_d(t)] + \alpha \tilde{x}(t) + \alpha^2 c_1(t) + \alpha c_1(t)}{2}}
\]

where \(\theta\) is an arbitrarily small positive number, \(0 < \gamma < 1\) and \(l_{1g} > 0\).

Proof: The Lyapunov function is defined as follows:

\[V_1(t) = \frac{1}{4}(e_1^T(t)e_1(t))^2\]  

(25)

The infinitesimal generator of \(V_1(x, t)\) for the system given by (25) satisfies

\[LV_1(t) = e_1^T(t)e_1(t)e_1^T(t)\frac{d}{dt}z(t) + \chi_1\]  

(26)

By using Young’s inequality to analyze the terms on the right-hand side of equation (26), we obtained the following expression:

\[e_1^T(t)e_1(t)e_1^T(t)\frac{d}{dt}z(t) \leq \frac{3}{4\mu_1^4} |e_1(t)|^4 + \frac{1}{4\mu_1^4} |z(t)|^4\]  

(27)

Next, we continuously defined an auxiliary control function \(\tilde{\chi}(\tilde{e}_1(t))\) as: \(\tilde{\chi}(\tilde{e}_1(t))_1 = -A_1 \text{Sig}(\tilde{e}_1(t))\gamma\) while considering that \(\tilde{e}_1(t)_{3x1} = [e_{11} e_{12} e_{13}]^T\) and \(A_1_{(3x3)} = \text{diag} \{l_{11}, l_{12}, l_{13}\}\).}

Then,

\[e_1^T(t)e_1(t)e_1^T(t)\frac{d}{dt}z(t) \leq -\frac{3}{4\mu_1^4} |e_1(t)|^4 - \frac{1}{4\mu_1^4} |z(t)|^4\]  

(28)
where \( \tilde{t}_1 = -2t_1, \ t_1 \equiv 2 \frac{x}{2} l_{1_{\text{min}}}, l_{1_{\text{min}}} = \min\{l_{1j}\}, \) and \( \delta = \frac{\gamma + 3}{4} \).

By substituting inequalities (27) and (28) into equation (26), then

\[
LV_1(t) \leq -\tilde{t}_1 V_1^i(t) + \alpha (c_1^T(t)c_1(t))^2 + \frac{3}{4} \mu_1^2 |c_1(t)|^2 + \frac{1}{4 \mu_1^2} |z(t)|^2 \tag{29}
\]

Further, we defined the Lyapunov function \( V_2(t) \) as follows:

\[
V_2(t) = \frac{1}{4} (z^T(t)z(t))^2 + V_1(t) \tag{30}
\]

The infinitesimal generator is

\[
LV_2(t) = z^T(t)z(t)z^T(t)[M_{s}^{-1}B(\theta)u(t) - \tilde{X}_d(t) - \alpha z(t) - \alpha \chi(c_1(t))]
- \alpha^2 c_1(t) - \chi(c_1(t))] + \frac{1}{2} Tr\{H^T B_1^T(\theta)M_{s}^{-1}[2z(t)z^T(t)
+ z^T(t)z(t)I]M_{s}^{-1}B_2(\theta)I\} + LV_1(t) \tag{31}
\]

where \( I \) denotes a suitably dimensioned identity matrix.

In addition, by considering the definition of the Frobenius norm, norm compatibility, and equation (8) while using Young’s inequality, we could derive the following expressions:

\[
\begin{align*}
\frac{1}{2} Tr\{H^T B_1^T(\theta)M_{s}^{-1}[2z(t)z^T(t) + z^T(t)z(t)I]M_{s}^{-1}B_2(\theta)I\}
& \leq \frac{3}{4} (z^T(t)z(t)) ||M_{s}^{-1}||^2_F ||B_2(\theta)I||^2_F
\leq \frac{3}{4} (z^T(t)z(t)) ||M_{s}^{-1}||^2_F (m + 3) ||B_2(\theta)I||^2_F
\leq \frac{3}{4} (z^T(t)z(t)) h \leq \frac{3}{4} \mu_2^2 |z(t)|^4 + \frac{1}{m^2} h^2
\end{align*}
\tag{32}
\]

By substituting the controllers (23) and (32) into (31), then

\[
LV_2(t) \leq (\alpha + \frac{3}{4} \mu_1^2) |c_1(t)|^4 - \tilde{t}_1 V_1^i(x, t) + z^T(t)z(t)z^T(t) \chi(z(t)) + \frac{1}{2 \mu_2^2} h^2 \tag{33}
\]

Likewise, an auxiliary control was defined as follows: \( \chi(z(t))_{3 \times 1} = -A_2 \text{Sig}(z(t))^\top \), where \( z(t)_{3 \times 1} = [z_{11} z_{12} z_{13}]^\top \) and \( A_2(3 \times 3) = \text{diag}\{l_{21}, l_{22}, l_{23}\} \).

Then,

\[
\begin{align*}
z^T(t)z(t)z^T(t) \chi(z(t)) & \leq -\sum_{j=1}^{3} |z_{1j}|^2 \cdot l_2 \cdot (\frac{1}{2} \sum_{j=1}^{3} |z_{1j}|^2) \frac{\gamma + 3}{4}
= -2l_2(\frac{1}{2} \sum_{j=1}^{3} |z_{1j}|^2)^2 \frac{\gamma + 3}{4}
= -\tilde{t}_2[(\frac{1}{2} \sum_{j=1}^{3} |z_{1j}|^2)^2]^{\delta}
\end{align*}
\tag{34}
\]

where \( \tilde{t}_2 = -2l_2, \ l_2 = 2 \frac{\gamma + 3}{4} l_{2\min}, \) and \( l_{2\min} = \min\{l_{2j}\}. \)

By substituting of equation (34) into equation (33), then

\[
LV_2(t) \leq -\tilde{t}_1 V_1^i(t) - \tilde{t}_2[(\frac{1}{2} \sum_{j=1}^{3} |z_{1j}|^2)^2]^{\delta} + \frac{1}{2 \mu_2^2} h^2
\leq -\tilde{V}_2^i(t) + d_c \tag{35}
\]
Here, $\tilde{l} = \min\{\tilde{l}_1, \tilde{l}_2\}$ and $d_c = \frac{1}{2\mu} \hat{h}^2$.

Furthermore, a scalar $0 < l_0 \leq 1$ exists such that (35) can be rewritten as follows:

$$LV_2(t) \leq -l_0 \tilde{l} V_2^2(t) - (1 - l_0) \tilde{l} V_2^2(t) + d_c$$  \hspace{1cm} (36)

Let $\Theta = \{ x \mid V_2^2(t) > (d_c/(1 - l_0)\tilde{l}) \}$, then

$$LV_2(t) \leq -l_0 \tilde{l} V_2^2(t)$$  \hspace{1cm} (37)

Then, by integrating the above inequality,

$$T(z(t)) \leq \frac{1}{l_0 l(1 - \delta)} [V_2^{1-\delta}(z_0) - (\frac{d_c}{(1 - l_0)\tilde{l}})^{\frac{1-\delta}{\delta}}]$$

Therefore, according to Lemma 1 and Lemma 2, the stochastic error system is finite-time stable. This completes the proof of Theorem 1.

Thus, we created a finite-time controller (23) that can perform stochastic tracking while handling random rehabilitee mass variations and velocity constraints. Fig. 3 shows a block diagram of the proposed ODW control system.

Remark 7: Controller (23) incorporates the constrained velocity $\dot{X}(t)$ and the fixed constant $m_s$ to improve user safety and the ODW’s tracking performance. This is a novel approach to handling velocity constraints for systems with stochastic dynamics.

Remark 8: Generally speaking, asymptotic stability implies that the trajectory of a system converges to equilibrium as time goes to infinity. Thus, extensive trajectory tracking errors may cause the ODW to collide with objects around it in the transient motion stage. The finite-time stability of dynamic systems may give rise to fast, transient, and high-precision performance, so the tracking controller (23) is vital to guarantee user safety.
Remark 9: In contrast to the previous methods achieving finite-time control [37,38], the developed controller (23) with velocity constraints can ensure practical finite-time stability for the stochastic rehabilitative training walker system, which is more applied to human-robot systems to improve the performances of tracking and safety.

5 Simulation results

As shown in this section, we evaluated the proposed tracking control algorithm by simulating an ODW. To study our algorithm’s performance, we assumed that the ODW follows a curved path and has the design parameters shown in Table 1. Table 2 provides the ODW’s physical parameters and the random rehabilitee mass variation used in the simulations. The initial values of the ODW system are listed in Table 3. Tables 4 and 5 list the velocity constraints and design parameters of the controller (23), respectively. The ODW’s path $X_d(t)$ is given as follows:

$$x_d(t) = 2\cos^3(0.1t)$$
$$y_d(t) = 2\sin^3(0.1t)$$
$$\theta_d(t) = \frac{\pi}{4}$$

Table 1 Design Parameters of the ODW

| $r_0$   | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $l_i$ |
|---------|--------------|--------------|--------------|--------------|-------|
| 0.16m   | $(L - r_0)m$ | $(L + r_0)m$ | $(L - r_0)m$ | $(L + r_0)m$ | $(\lambda_i / \cos(\theta_i - \varphi_i))m$ |

Table 2 Physical Parameters of the ODW

| $M$ | $m$ | $L$ | $I_0$ |
|-----|-----|-----|-------|
| 58kg | 80$(1 + \text{rand}())$kg | 0.4m | 27.7kg·m$^2$ |

Table 3 Initial System State

| $x(0)$ | $y(0)$ | $\theta(0)$ | $\dot{x}(0)$ | $\dot{y}(0)$ | $\dot{\theta}(0)$ | $v_1(0)$ | $v_2(0)$ | $v_3(0)$ | $v_4(0)$ |
|--------|--------|-------------|-------------|-------------|-----------------|----------|----------|----------|----------|
| 2.1m   | 0.1m   | 0.75rad     | 0.0m/s      | 0.0m/s      | 0.0m/s          | 0.0m/s   | 0.0m/s   | 0.0m/s   | 0.0m/s   |

The simulation results are given in the following figures.
Table 4  Velocity Constraints

| $v_x(t)$ | $v_y(t)$ | $v_\theta(t)$ |
|----------|----------|--------------|
| 0.25 m/s | 0.25 m/s | 0.15 rad/s |

Table 5  Controller Design Parameters for Simulation 1

| $\mu_1$ | $\mu_2$ | $m_s$ | $\alpha$ | $\gamma$ | $Q$ | $R$ | $A_1$ | $A_2$ | $\Delta g$ |
|---------|---------|-------|----------|---------|-----|-----|-------|-------|----------|
| 1.5     | 7.0     | 70    | -17.5    | 0.91    | diag{18, 20, 20} | diag{10, 15, 10} | diag{1, 1, 0.1} | diag{1, 1, 0.1} | 0.01     |

Figs. 4-6 plot the tracking performance for the $x$-axis, $y$-axis, and orientation angle, respectively. Fig. 7 shows how well the ODW tracked the curved path, and Fig. 8 shows the trajectory tracking errors. These figures indicate that the controller could handle random rehabilitee mass variations that would otherwise have reduced the robot’s tracking performance. In addition, the tracking error system given by (22) achieved finite-time stability. Thus, the s-
Fig. 8  Trajectory tracking errors

Fig. 9  Motion velocity (x-axis)

Fig. 10  Motion velocity (y-axis)

Fig. 11  Motion velocity (orientation angle)

Fig. 12  Control inputs of $f_1$ and $f_2$

Fig. 13  Control inputs of $f_3$ and $f_4$
The stochastic ODW model (7) is clearly effective in dealing with random parameter variations using controller (23).

Figs. 9-11 show the ODW’s velocities with regard to the $x$-axis, $y$-axis, and orientation angle, respectively, demonstrating that they remained within the prescribed range. Thus, our novel velocity constraint technique, which used the model predictive approach given by equations (11) and (12) in designing the tracking controller (23), is effective in limiting velocity changes. This method has the advantage that the velocity is constrained by combining the models of the ODW’s kinematics and dynamics. The control input curves are shown in Figs. 12-13, and the influence of the random mass change of different users (Fig. 14) on the tracking performance was effectively suppressed under the used control action.

To investigate the effect of the different design parameters on the tracking error, we considered a different set of controller parameters. Here, the trajectory followed the same curved path as before, as shown by equation (38). Also, the system parameters, ODW physical parameters, initial state, and velocity constraints were also the same (Tables 1-4). However, the controller’s design parameters were different (Table 6). Figs. 15-21 show the results of this simulation experiment.

| Table 6 | Controller Design Parameters for Simulation 2 |
|---------|---------------------------------------------|
| $\mu_1$ | 1.8 | $\mu_2$ | 6.9 | $m_s$ | 70 | $\alpha$ | $-6.6$ | $\gamma$ | 0.95 |
| $Q$     | $diag\{30, 5, 35\}$ | $R$     | $diag\{1, 10, 3\}$ | $A_1$ | $diag\{0.1, 0.1, 0.1\}$ | $A_2$ | $diag\{0.1, 0.1, 0.1\}$ | $\Delta_0$ | 0.01 |
Fig. 15 Tracking of the curved path

Fig. 16 Motion velocity (x-axis)

Fig. 17 Motion velocity (y-axis)

Fig. 18 Motion velocity (orientation angle)

Fig. 19 Control inputs of $f_1$ and $f_2$

Fig. 20 Control inputs of $f_3$ and $f_4$
Fig. 21 User’s random mass

Fig. 15 shows the curve path tracking, demonstrating that the used different design parameters still enabled the controller to achieve accurate path tracking. Figs. 16-18 plot the ODW’s actual velocity in the $x$-axis, $y$-axis, and orientation angle, respectively. Again, it is evident that the used velocity constraint technique in designing the tracking controller (23) is effective in preventing the velocity from going beyond a predetermined safe range. The control input curves are given in Figs. 19-20, and the user’s random mass was plotted, as shown in Fig. 21. It can also be seen that the design parameters affected both the tracking performance and constrained velocity, so they should thus be carefully adjusted.

Next, we conducted a simulation experiment to compare our results with those of [32] and further evaluated the effectiveness of the proposed finite-time tracking control method. The model proposed in [32] handles random changes in the parameters related to center of gravity, but it does not consider the impact of having different rehabilitees in a closed system, even though this seriously affects tracking. This control method also does not constrain velocity, meaning that users may be in danger during rehabilitation if the ODW’s velocity suddenly changes due to unpredictable environment effects. Here, we used the same parameter values as in [32] and retained the same curved path as before but applied the controller in [32] to the ODW, as described by our stochastic model (7), and the results are shown in Figs. 22-28.
Figs. 22-24 plot the ODW tracking performance with regard to the $x$-axis, $y$-axis, orientation angle, respectively, while Fig. 25 shows how well the controller tracked the curved path. This makes it clear that the controller proposed in [32] cannot achieve finite-time stable tracking in the presence of randomly varying user masses. In particular, Fig. 25 shows that the rehabilitee could have been in danger because the actual path is a long way from the specified path. Figs. 26-28 show the ODW’s velocity in the $x$-axis, $y$-axis, and orientation angle, respectively. Here, it can be seen that the velocity was not constrained to within any given range and at times significantly exceed the limiting values. Again, this could have hurt a rehabilitee because the user and ODW would not have moved in harmony. These results suggest that the controller proposed in [32] cannot deal with random user mass variations or enforce velocity constraints. Clearly, actual human-robot systems should be usable by a wide range of different rehabilitees, and their velocities should be kept within safe limits. This is why we felt it is important to design a finite-time
controller that can handle different masses and limit the velocities involved so that human-robot systems can provide safe rehabilitation environments.

To further demonstrate the advantages of our proposed finite-time tracking controller, we applied our controller (23) with the design parameters given in Table 6 and a constant center-of-gravity offset of \( r_0 = 0.16m \) to the ODW model given in [32], while the other parameters and motion trajectory remained the same. The results were demonstrated as shown below.
Figs. 29-31 show the ODW tracking performance with regard to the $x$-axis, $y$-axis, and orientation angle, respectively, while Fig. 32 shows how well the controller tracked the linear path. These figures indicate that our proposed controller (23) can achieve safe and finite-time accurate tracking based on the model used in [32]. In addition, Figs. 33-35 show that the ODW’s velocities remained within the prescribed range. The approach presented in [32] only considers random shifts in the center of gravity and does not attempt to ensure safe ODW velocities, meaning that their controller more conservatively deals with velocity constraints. Sudden changes in the ODW’s velocity can thus place a rehabilitee in danger, where the robot’s movement can exceed what the user can deal with. Thus, it can be seen that our tracking controller (23), which constrains velocity and finite-time tracking, is more universal than the one in [32] and that it can enable rehabilitees to train safely at manageable velocities.
6 Experimental results

To verify the effectiveness of the stochastic finite-time velocity constraint control method proposed in this paper, a trajectory tracking experiment of actual rehabilitation training using ODW was performed. In the experiment, two graduate students participated, where one was a walking trainer, and the other recorded the experimental data.

The trainer selected the walk mode within the ODW and placed the arms on the handrail to support the body weight. The ODW reads the control instructions through the serial port and returns the motion state information to the controller, and the controller calculates the error between the actual motion state and specified motion state, thus continuously adjusting the motion speed and the motion trajectory of the ODW according to the calculating error, restricting the motion speed in a specified range, and enabling the ODW to realize tracking of the specified motion trajectory. The experimental environment is shown in Fig. 36.
The reference trajectory \( X_d(t) \), physical parameters, velocity constraints, and controller are the same as those used in Section 5. The experimental results are given below.
Figs. 37-40 show the tracking performance for the $x$ position, $y$ position, orientation angle, and path tracking, respectively. These results show that the...
ODW can help randomly changing users in achieving stable tracking motion in finite time. Although there were some tracking angle errors at the beginning, indicating that the robot-human motion environment has a great impact on angle tracking, the controller quickly restrained the environment changes, and the robot gradually realized tracking training.

The actual velocity curves of the ODW are shown in Figs. 41-43, from which it can be seen that the velocity of each axis is limited within a specified range, indicating that the velocity constraint method proposed in this paper not only ensures safe velocities but also provides a velocity constraint method for the random ODW system.

These experimental results confirm that the proposed finite-time stochastic controller effectively realizes stable tracking training while properly constraining the actual motion velocity of the ODW and suppressing the impact of have various users on the tracking performance. Therefore, the ODW with the proposed control method with velocity constraints is reliable and can efficiently be used in assisting rehabilitees in walking training.

7 Conclusions

In this study, we proposed a stochastic dynamic model for an ODW that can allow for random mass variations among rehabilitees. Also, we adopted a novel design approach that restricts the ODW’s velocity by controlling the velocity input to each omniwheel. In addition, we used a standard Lyapunov function to design a tracking controller for the stochastic ODW system to ensure finite-time tracking. While designing the proposed method, we mainly focused on constraining velocity and dealing with random rehabilitee mass variations, as both issues have not been investigated in previous studies. A simulation was conducted, and the experimental results demonstrated that the ODW can provide safe tracking during training. Also, in addition to the ODW, the proposed scheme can also be applied to other human-robot devices.

Acknowledgements

This work was supported by the JSPS KAKENHI under Grant 15H03951, the CANON Foundation, the CASIO Science Promotion Foundation, the Liaoning Natural Science Foundation of China under Grant 2019ZD0203, and Basic research program of Education Department Foundation of Liaoning Province under Grant LJGD2019017.

Declarations

No potential conflict of interest was reported by authors.
References

1. Dong, M. D., Zhou, Y., Li, J. F., et al.: State of the art in parallel ankle rehabilitation robot: a systematic review. Journal of Neuroengineering and Rehabilitation. DOI: https://doi.org/10.1186/s12984-021-00845-z.(2021)

2. Miao, Q., Zhang, M. M., Wang, C. Z., et al.: Towards optimal platform-based robot design for ankle rehabilitation: the state of the art and future prospects. Journal of Healthcare Engineering. 1-9(2018)

3. Berger, A., Horst, F., Muller, S., et al.: Current state and future prospects of EEG and fNIRS in robot-assisted gait rehabilitation: a brief review. Frontiers in Human Neuroscience. 13, 1-17(2019).

4. Miao, M. D., Gao, X. S., Zhu, W.: A construction method of lower limb rehabilitation robot with remote control system. Applied Sciences. https://doi.org/10.3390/app11020867.(2021)

5. Yang, Y. F., Kong, X. Z., Yang, J., Li, G. L., et al.: Modeling and simulation of an unpowered lower extremity exoskeleton based on gait energy. Mathematical Problems in Engineering. 1-15(2020)

6. Shi, D., Zhang, W. X., Zhang, W., et al.: A review on lower limb rehabilitation exoskeleton robot. Chinese Journal of Mechanical Engineering. 32, 1-11(2019)

7. Abooee, A., Arefi, M. M., Sedghi, F., Abootalebi, V.: Robust nonlinear control schemes for finite-time tracking objective of a 5-DOF robotic exoskeleton. International Journal of Control. 01-37(2018)

8. Cao, J., Xie, S. Q., Das, R.: MIMO sliding mode controller for gait exoskeleton driven by pneumatic muscles. IEEE Transactions on Control Systems Technology. 26(1),274-281(2017)

9. Sun, P., Wang, S. Y.: Redundant input safety tracking for omnidirectional rehabilitative training walker with control constraints. Asian Journal of Control. 19(1), 116-130(2017)

10. Benzaoui, M., Chekireb, H., Tadjine, M.: Trajectory tracking with obstacle avoidance of redundant manipulator based on fuzzy inference systems. Neurocomputing.196, 23-30(2016)

11. Nagamani, G., Joo, Y. H., Soundalarajaraj, G., Mohajerpoor, R.: Robust event-triggered reliable control for T–S fuzzy uncertain uncertain systems via weighted based inequality. Information Science. 512, 31-49(2020)

12. Hu, S. F., Jie, Z., Chen, C.: State estimation for dynamic systems with unknown process inputs and applications. IEEE Access. 6, 14857-14869(2018)

13. Liu, Z. G., Wu, Y. Q.: Modelling and adaptive tracking control for flexible joint robot with random noises. International Journal of Control. 87(12), 2499-2510(2014)

14. Cui, M. Y., Xie, X. J., Wu, J. Z.: Dynamics modeling and tracking control of robot manipulators in random vibration environment. IEEE Transactions on Automatic Control. 58(6), 1540-1545(2013)

15. Cui, M. Y., Wu, Z. J., Xie, X. J.: Output feedback tracking control of stochastic Lagrangian systems and its application. Automatica. 50(5), 1424-1433(2014)

16. Duecker, D. A., Geist, A. R., Kreuzer, E., Solowjow, E.: Learning environmental field exploration with computationally constrained underwater robots: Gaussian processes meet stochastic optimal control. Sensor. 19, 1-28(2019)

17. Alsulabaie, M. A., Rogers, E.: Robustness and load disturbance conditions for state based iterative leaning control. Optimal Control Applications and Methods. 39(6): 1965-1975(2018)

18. Sun, P., Wang, S. Y., Chang, H. B.: Tracking control and identification of interaction forces for a rehabilitative training walker whose center of gravity randomly shifts. International Journal of Control. 6: 1-21(2019)

19. Meng, W., Liu, Q., Zhou, Z.: Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation. Mechatronics. 31, 132-145(2015)

20. Ahmadi, S. M., Fateh, M. M., Robust control of electrically driven robot using adaptive uncertainty estimation. Computers and Electrical Engineering. 56, 674-687(2016)

21. Hwang, B., Oh, B. M., Jeon, D.: An optimal method of training the specific lower limb muscle group using an exoskeletal robot. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 26(4), 830-838(2018)
22. Chen, X. L., Zhao, H., Zhen, S. C.: Adaptive robust control for a lower limbs rehabilitation robot running under passive training mode. IEEE/CAA Journal of Automatica Sinica. 6(2), 493-502 (2019)

23. Alsubaie, M., Rogers, E.: Robustness and load disturbance conditions for state based iterative learning control. Optimal control applications and methods. 39(6), 1965-1975(2018)

24. Lu, L., Yao, B.: A performance oriented multi-loop constrained adaptive robust tracking control of one-degree-of-freedom mechanical systems: Theory and experiments. Automatica. 50(4), 1143-1150(2014)

25. Sun, T. R., Pan, Y. P.: Robust adaptive control for prescribed performance tracking of constrained uncertain nonlinear systems. Journal of the Franklin Institute. 356(1), 18-30(2019)

26. Sun, P., Wang, S. Y.: Guaranteed cost non-fragile tracking control for omnidirectional rehabilitative training walker with velocity constraints. International Journal Control, Automation and Systems. 14(5), 1340-1351(2016)

27. Karl, W., Mohamed, W. M.: Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. IEEE Transactions on Control System Technology. 24(4), 1394-1406(2016)

28. Xiao, H. Z., Li, Z. J., Yang, C. G.: Robust stabilization of a wheeled mobile robot using predictive control based on neurodynamics optimization. IEEE Transactions on Industrial Electronics. 64(1), 505-516(2017)

29. Cao, Q., Sun, Z. Q., Xia, Y. Q., and Dai, L.: Self-triggered MPC for trajectory tracking of unicycle-type robots with external disturbance. Journal of the Franklin Institute. 356(11), 5592-5610(2019)

30. Henriksson, E., Quevedo, D. E., Peters, E. Q. W.: Multiple-loop self-triggered model predictive control for network scheduling and control. IEEE Transactions on Control Systems Technology. 23(6): 2167-2181(2015)

31. Sun, P., Wang, S. Y.: Redundant input guaranteed cost non-fragile tracking control for omnidirectional rehabilitative training walker. International Journal of Control, Automation and Systems 13(2), 454-462(2015)

32. Chang, H. B., Sun, P., Wang, S. Y.: Output tracking control for omnidirectional rehabilitative training walker with incomplete measurements and random parameters. International Journal of System Science. 48(12), 2509-2521(2017)

33. Cui, M. Y., Wu, Z. J., Xie, X. J., and Shi, P.: Modeling and Adaptive Tracking for a Class of Stochastic Lagrangian Control Systems. Automatica. 49, 770-779(2013)

34. Khas’minskii, R. Z.: Stochastic stability of differential equations (2nd ed.). Berlin: Springer-Verlag(2012)

35. Sun, P., Wang, S. Y.: Redundant input guaranteed cost switched tracking control for omnidirectional rehabilitative training walker. International Journal of Innovative Computing Information and Control. 10(3), 883-895(2014)

36. Zhai, D. H., Xia, Y. Q.: Adaptive finite-time control for nonlinear teleoperation systems with asymmetric time-varying delays. International Journal of Robust and Nonlinear Control. 26, 2586-2607(2016)

37. Shen, G. H., Xia, Y. Q., Zhang, J. H., Cui, B.: Adaptive fixed-time trajectory tracking control for mars entry vehicle. Nonlinear Dynamics. DOI: 10.1007/s11071-020-06088-2(2020)

38. Zhang, C., Li, Y., Qi, G., et al: Distributed finite-time control for coordinated circumnavigation with multiple non-holonomic robots. Nonlinear Dynamics. 98(2), 573-588(2019)