IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Systematic design of the time-independent and computable controller based on zero-division-avoidable smoother for a desired orbit in phase space

Ken Masuya[1].

Abstract— **This letter proposes a method to systematically design a time-independent controller for a desired orbit in phase space. A time-independent controller is essential in robots that physically interact with humans or the environment. An approach to designing such a controller is based on the virtual dynamics of the desired orbit (VDDO), in which the desired orbit is assumed as a constraint. However, depending on the desired orbit, zero-division happens, and then the computation of control input breaks down. To address this issue, a zero-division-avoidable smoother, which functions as a low-pass filter and maintains computability even when the computation includes zero-division, is applied to compute the controller input based on the VDDO. This application establishes a systematic design of a VDDO-based controller that avoids zero-division. We investigated the performance of the proposed controller via experiments and simulations for three given orbits: a unit circle, super-ellipse, and spiric section. Results showed that the proposed time-independent controller can avoid zero-division while approaching the desired orbits. Furthermore, an experiment in which a human forces a robot to stop showed that the robot could restart from an unfavorable state and approach the desired orbits once more.**

I. INTRODUCTION

The rising demand for robots working in living environments has enhanced the opportunity for robots to physically interact with humans or environments. For such robots, time-dependent motion control is unsuitable because unpredicted interactions cannot easily be predetermined, thus necessitating time-independent motion control.

Thus far, several time-independent motion controllers for robots exhibiting physical interaction have been developed. In particular, periodic or quasi-periodic motions have been investigated to address human or legged robot motions, such as walking and swinging. These motions are represented as a desired orbit, and this letter designs a time-independent controller that converges to the desired orbit representing periodic or quasi-periodic motion. One method for designing time-independent controllers for such motions is the use of a preset state/output relationship that converges to the desired orbit [1], [2], [3], [4], [5]. Although this method is applicable if a suitable output is preobtained, such an output cannot easily be predetermined. Another method is to correspond the controller to be designed to an existing limit cycle. This approach is widely employed in robotics [6], [7], [8], [9], [11], [12], [13]; however, most of those studies are limited to periodic motion. Even when the desired orbit can be extended to quasi-periodic motion, incorrect handling of singularities when mapping the controller to the limit cycle may cause computation failure. Other method involves generating a vector field based on criteria such as the energy field [10] or distance from the desired orbit [14], [15], [16], [17], [18], [19]. However, for the former criteria, an energy field cannot easily be obtained, depending on the given orbit. Moreover, for the latter criteria, it is challenging to choose an appropriate point on the desired orbit when multiple nearest points are available. This dependency on the desired orbit can be relaxed by considering a virtual constraint [20], [21]. However, the issue of avoiding computation failure due to improper handling of singularities remains to be addressed.

To address this issue, this study designed a time-independent controller that avoids zero-division. Hereinafter, for ease, we consider a simple 1-DOF system with a revolute joint and assume that the desired orbit is in a phase space consisting of its angle and angular velocity. To this end, we first focus on the virtual constraint approach [20], [21]. In this approach, the control input is designed based on the constraint corresponding to the desired orbit. Considering the constraint stabilization by Baumgarte [22], this approach can also be interpreted as setting the virtual dynamics of the desired orbit (VDDO). Hereinafter, this controller is termed the VDDO-based controller. However, as aforementioned, this controller input still suffers from computation failure, depending on the desired orbit. In particular, as zero-division causes a fatal event for robotic motion, it should be avoided. L2 regularization is often used to avoid zero-division. However, its input is likely to become zero as the state approaches the conditions with zero-division, and then the robot cannot restart alone after stopping at these conditions once.

Previously, the author proposed a filter that functions as a low-pass filter (LPF) and maintains computability even when the computation includes zero-division [5], [23]. Hereinafter, we termed this filter a zero-division-avoidable (ZDA) smoother based on its function. In this method, because the LPF is obtained from the optimization of a specific evaluation function, the calculations that may cause zero-division were reconstructed into this function. Using this idea, the center of rotation of the robot foot or the knee joint was estimated in the previous studies [5], [23].

In this letter, we apply this idea to the design of the VDDO-based controller. The computation of the control

*This work was supported by JSPS KAKENHI under Grant JP23K13292. (Corresponding author:Ken Masuya.)

[1]Ken Masuya is with Faculty of Engineering, University of Miyazaki, Miyazaki 889-2192, Japan e-mail: ken.masuya@ieee.org

input is reconstructed into an evaluation function, and we can systematically obtain the control input that avoids zero-division by optimizing the function. That is, this letter's main contribution is to provide a systematic design of the VDDO-based controller that avoids zero-division.

The remainder of this letter is as follows: Section II presents the design of the VDDO-based controller for a 1-DOF robot. In Section III, we propose a systematic design of the VDDO-based controller that can avoid zero-division. Through simulations for three types of closed orbit, we compare the proposed controllers with VDDO-based controllers in Section IV. Then, the performance of the proposed controller is experimentally demonstrated in Section V. Additionally, we verify the behavior when human interaction occurs. Finally, Section VI presents the conclusions and future works.

II. Controller design based on the virtual dynamics of closed trajectory

A. Simple 1-DOF system

In this letter, we deal with a simple 1-DOF system such as a robot with one revolute joint, as shown in Fig. 1. Let θ and ω be the joint angle and angular velocity, respectively, its ideal state-space representation can be written as follows:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ u \end{bmatrix}, \tag{1}$$

where $\boldsymbol{q} = [\theta\ \omega]^{\mathrm{T}}$ is the state of the robot, and u is the input angular acceleration.

B. VDDO-based controller

This letter aims to design a time-independent controller for the system (1) that approaches a desired orbit in the phase space. Let a formula $\xi(\boldsymbol{q}) = 0$ be a desired orbit in the phase space, and $\xi(\boldsymbol{q})$ is assumed as a dimensionless quantity. A controller that converges to $\xi(\boldsymbol{q}) = 0$, can be designed based on the VDDO as follows:

$$\dot{\xi} = -K(\boldsymbol{q})\xi, \tag{2}$$

where $K(\boldsymbol{q}) \geq 0$ is the controller gain. Considering the system (1), the left-hand side of (2) can be rewritten as

$$\begin{aligned} \dot{\xi} &= \xi_\theta \dot{\theta} + \xi_\omega \dot{\omega} \\ &= \xi_\theta \omega + \xi_\omega u, \end{aligned} \tag{3}$$

where $\xi_\theta = \frac{\partial \xi}{\partial \theta}$, and $\xi_\omega = \frac{\partial \xi}{\partial \omega}$. By substituting (3) into (2), a control input can be obtained as follows:

$$u = -\xi_\omega^{-1}\big(\xi_\theta \omega + K(\boldsymbol{q})\xi\big). \tag{4}$$

(4) indicates that the state $\boldsymbol{q}$ approaches $\xi = 0$ as long as $K(\boldsymbol{q}) > 0$. This idea is also termed the virtual constraint approach for orbital stabilization [20], [21] and is similar to the constraint stabilization by Baumgarte [22]. As discussed in Section IV, the modified van der Pol equation [24] can also be derived from this idea. However, in practical cases, ξ_ω^{-1} causes zero-division, and thus, the robot may become out of control. Therefore, a design of the time-independent controller that avoids zero-division is necessary.

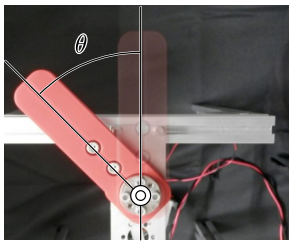


Fig. 1. Simple 1-DOF robot dealt with in this letter.

C. VDDO-based controller with L2 regularization

Introducing an L2 regularization term can prevent such a situation. In this design, (2) is assumed to have an error, and this error is minimized while minimizing the L2 norm of the input. From this viewpoint, the evaluation function is set as follows:

$$E = \frac{1}{2}\Big(\xi_\omega u + \xi_\theta \omega + K(\boldsymbol{q})\xi\Big)^2 + \frac{1}{2} w_u u^2, \tag{5}$$

where w_u is the weight of the input regularization term. The stationary condition of (5) for u leads the control input as follows:

$$u = -\frac{\xi_\omega}{w_u + \xi_\omega^2}\big(\xi_\theta \omega + K(\boldsymbol{q})\xi\big). \tag{6}$$

(6) indicates that the control input is always computable, including when $\xi_\omega = 0$. However, once the robot stops at the state where $\xi_\omega = 0$, restarting the motion autonomously is difficult because of zero input. This situation can occur when the robot physically interacts with its surroundings.

III. Proposed design of the computable and restartable VDDO controller

To design a computable and restartable VDDO controller, we apply the ZDA smoother to the VDDO-based controller. The ZDA smoother is detailed in Appendix, and the concept of the proposed design is illustrated in Fig. 2. In this design, we consider the following evaluation function instead of (5).

$$\begin{aligned} E = &\frac{1}{2}\Big(\xi_{\omega,k} u_k + \xi_{\theta,k}\omega_k + K(\boldsymbol{q}_k)\xi_k\Big)^2 \\ &+ \frac{1}{2} w_F \Big(u_k - u_{k-1}\Big)^2, \end{aligned} \tag{7}$$

where the subscript k denotes the variable at the time $k\Delta T$, ΔT is the sampling interval, and w_F is the weight of the second term. By considering the stationary condition of (7) for u_k, the control input is obtained as follows:

$$\begin{aligned} u_k = &\frac{w_F}{w_F + \xi_{\omega,k}^2} u_{k-1} \\ &- \frac{\xi_{\omega,k}}{w_F + \xi_{\omega,k}^2}\big(\xi_{\theta,k}\omega_k + K(\boldsymbol{q}_k)\xi_k\big). \end{aligned} \tag{8}$$

Similar to (6), (8) indicates that the control input is always computable even if $\xi_\omega = 0$. By contrast, the previous input u_{k-1} is maintained when $\xi_\omega = 0$ and helps restart the motion autonomously. Additionally, (8) implies that the initial input u_0 is allowable in the proposed controller. We investigate

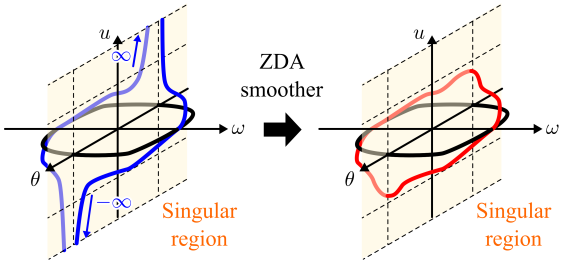


Fig. 2. Concept of the proposed design of the computable and restartable VDDO-based controller. To prevent the input divergence due to zero-division, we apply the ZDA smoother into the VDDO-based controller.

TABLE I

SUMMARY OF DIFFERENCES BETWEEN CONTROLLERS.

	Initial input u_0 [rad/s^2]	Previous input u_{k-1}	Weight for regularization [s^2]
P-SW	50.0	Available	0.01
P-LW	50.0	Available	1.0
P-WO	0.0	Available	0.01
VDDO	Not available	Not available	0.0
VDDO-IR	Not available	Not available	0.01

the effect of u_0 on the control performance in the simulation described below.

IV. SIMULATION RESULTS

A. Simulation setup

We investigate the performance of the proposed controller through simulations for three orbits: a unit circle, super-ellipse, and spiric section. As real applications require the different ranges of the angle θ and angular velocity ω, we considered the normalized angle $\bar{\theta}$ and angular velocity $\bar{\omega}$. That is,

$$\bar{\boldsymbol{q}} = \begin{bmatrix} \bar{\theta} \\ \bar{\omega} \end{bmatrix} = \begin{bmatrix} \alpha_\theta(\theta - \theta_c) \\ \alpha_\omega(\omega - \omega_c) \end{bmatrix}, \tag{9}$$

where α_θ and α_ω are the positive normalized factors for θ and ω, respectively. We set $\alpha_\theta = 1.0$ rad^{-1} and $\alpha_\omega = 0.2$ s/rad in the all evaluations below. Moreover, θ_c and ω_c denote the center of the motion range of θ and ω, respectively. In the below evaluation, for ease, we set $\theta_c = 0$ rad and $\omega_c = 0$ rad/s. To evaluate the effect of the initial state $\bar{\boldsymbol{q}}_0 = [\bar{\theta}_0\ \bar{\omega}_0]^{\mathrm{T}}$, we assigned several initial states to the simulator. Herein, both $\bar{\theta}_0$ and $\bar{\omega}_0$ were set from -2 to 2 every 1. In all simulations, the sampling interval ΔT was 2 ms, and the end time t_f was set to 5 s.

For each orbit, we compared the following five VDDO-based controllers:

- **P-SW** Proposed VDDO-based controller with initial input and small weight ($u_0 = 50.0$ rad/s^2 and $w_F = 0.01$ s^2)
- **P-LW** Proposed VDDO-based controller with initial input and large weight ($u_0 = 50.0$ rad/s^2 and $w_F = 1.0$ s^2)
- **P-WO** Proposed VDDO-based controller without initial input and with small weight ($u_0 = 0.0$ rad/s^2 and $w_F = 0.01$ s^2)
- **VDDO** VDDO-based controller designed by (4)
- **VDDO-IR** VDDO-based controller with input regularization (6) and $w_u = 0.01$ s^2

Table I summarizes the differences between these controllers.

To qualitatively evaluate the orbit error ξ, we employed the root-mean-square error (RMSE). In this letter, the RMSE was computed from ξ of the last 2 s as follows:

$$\mathrm{e}_* = \sqrt{\frac{1}{N_*}\sum_{i\in\mathcal{A}_*}\left(\frac{\Delta T}{2}\sum_{k\Delta T=t_f-2}^{t_f}\xi^2(\boldsymbol{q}_{k,i})\right)} \quad (* = \mathrm{W},\ \mathrm{WO}), \tag{10}$$

where $\boldsymbol{q}_{k,i}$ is the state at the time $k\Delta T$ in the i-th locus. $\mathcal{A}_\mathrm{W}$ and $\mathcal{A}_\mathrm{WO}$ are the sets of trials including and not including the computable trials that the state does not move from the initial state, respectively, and both sets include the computable results moving from the initial state. N_W and N_WO are the numbers of trials in $\mathcal{A}_\mathrm{W}$ and $\mathcal{A}_\mathrm{WO}$, respectively.

B. Simulation result 1: Unit circle

First, we investigate the control performance when the desired orbit is drawn as a unit circle represented as

$$\xi = \bar{\theta}^2 + \bar{\omega}^2 - 1. \tag{11}$$

As $\xi_\theta = 2\alpha_\theta\bar{\theta} = 2\alpha_\theta^2\theta$ and $\xi_\omega = 2\alpha_\omega\bar{\omega} = 2\alpha_\omega^2\omega$, a controller (4) is designed as

$$\begin{aligned} u &= -\frac{1}{2\alpha_\omega^2\omega}\left(2\alpha_\theta^2\theta\omega + K(\boldsymbol{q})\left(\alpha_\theta^2\theta^2 + \alpha_\omega^2\omega^2 - 1\right)\right) \\ &= -\frac{\alpha_\theta^2}{\alpha_\omega^2}\theta - \frac{K(\boldsymbol{q})}{2\alpha_\omega^2\omega}\left(\alpha_\theta^2\theta^2 + \alpha_\omega^2\omega^2 - 1\right), \end{aligned} \tag{12}$$

Although zero-division clearly occurs in (12) when $\omega = 0$, it can be avoided by setting the gain as follows:

$$K(\boldsymbol{q}) = K_\omega(\alpha_\omega\omega)^2 \geq 0, \tag{13}$$

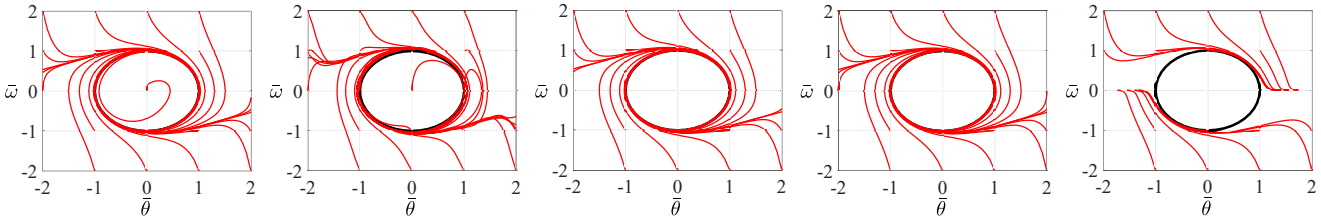
where $K_\omega > 0$ is the gain parameter. By using the gain design (13), the controller (12) can be rewritten as follows:

$$u = -\frac{\alpha_\theta^2}{\alpha_\omega^2}\theta - \frac{1}{2}K_\omega\omega\left(\alpha_\theta^2\theta^2 + \alpha_\omega^2\omega^2 - 1\right). \tag{14}$$

Note that the controller (14) corresponds to the modified van der Pol equation [24].

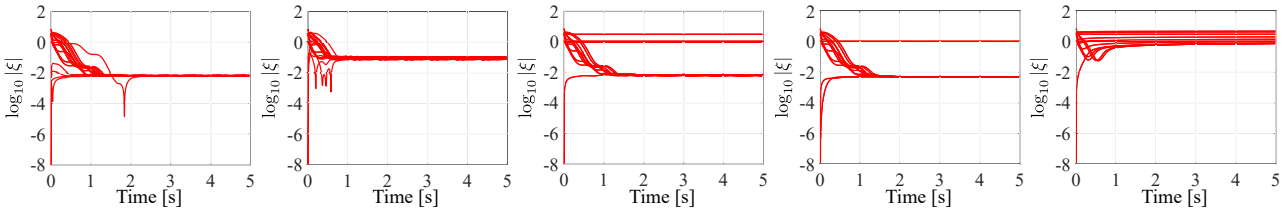
We conducted the simulation using the condition $K_\omega = 10.0$ s^{-1}. Herein, we selected the gain such that P-SW converges to the desired orbit around 1 s. Moreover, this policy was applied to the other simulations. The resultant phase portraits are shown in Fig. 3, and the time variation in error is shown in Fig. 4. In both figures, red lines represent the trajectories that are computable in a 5-s simulation. Table II presents the RMSEs of each controller in this case.

As illustrated in both figures, zero-division does not occur in all controllers owing to (13). However, Fig. 3(e) shows that the trajectories of VDDO-IR are trapped and stopped in a region satisfying $\xi_\omega = 0$; namely, $\omega = 0$ rad/s because the input becomes smaller as the state approaches the abovementioned region. Due to this trapping, both RMSEs become larger as shown in Table II. Controllers, except for



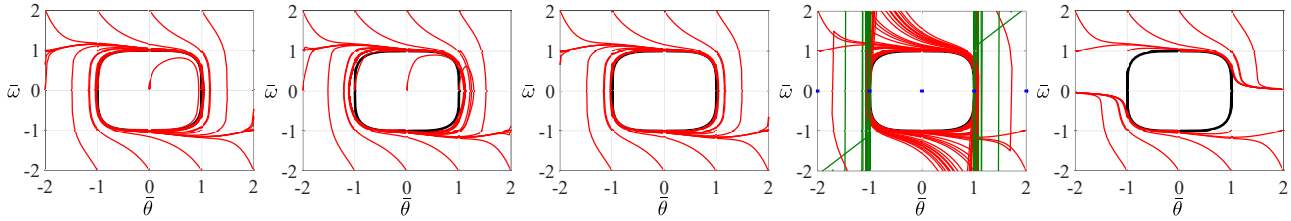
(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 3. Simulation results of phase portraits for a desired orbit drawn as a unit circle. Red lines represent the results when the input u is computable in the simulation period, and the black line represents the desired trajectory.



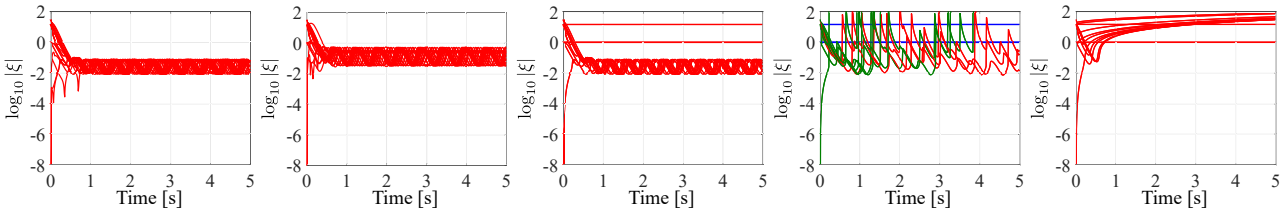
(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 4. Simulation results of orbit error represented by $|\xi|$ in the case of a unit circle.



(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 5. Simulation results of phase portraits for a desired orbit drawn as super-ellipse. Red lines represent the results when the input u is computable in the simulation period, and the black line represents the desired orbit. Blue lines represent the results when the input is incomputable from the initial state. Green lines represent the results that the input becomes incomputable during the simulation.



(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 6. Simulation results of orbit error in the case of a super-ellipse. Red lines represent the results when the input is computable in the simulation period. Blue lines represent the results that the input is incomputable from the initial state and the values in the graph are obtained from those initial conditions. Green lines represent the results that the input becomes incomputable during simulation.

VDDO-IR, can converge to a state on a unit circle (11). Moreover, the trajectories of those controllers are similar.

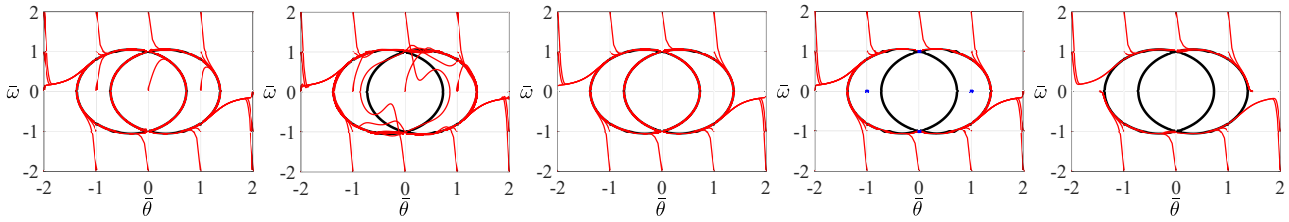
Meanwhile, as illustrated in Fig. 3(d), the result of VDDO has no trajectory from the initial state $\bar{\boldsymbol{q}}_0 = \mathbf{0} = [0\ 0]^{\mathrm{T}}$. Similarly, as shown in Fig. 3(c), P-WO, the proposed controller without u_0, does not move from the initial state with $\bar{\omega}_0 = 0$. This immobility accounts for the differences between e_{W} and e_{WO} of these controllers in Table II. Compared with these results, the proposed controller with u_0 can converge to the desired orbit, even in the above cases; consequently e_{W} and e_{WO} have the same values.

Comparing Fig. 4(a) with Fig. 4(b), the error of P-LW is larger than that of P-SW because the previous input u_{k-1} is prioritized by the large w_F even if $\xi_\omega \neq 0$. Additionally, the error e_{W} of P-SW is almost the same as that of VDDO. Therefore, for the unit circle, it is validated that the proposed controller with an initial input and small weight can converge to the desired orbit independent of the initial state.

C. Simulation result 2: Super-ellipse

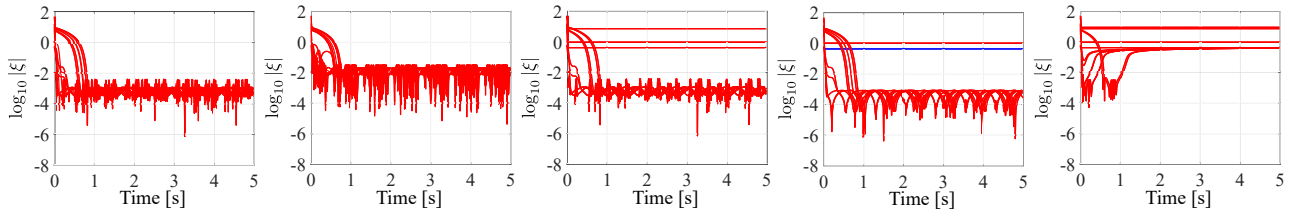
We compared the abovementioned controllers for a super-ellipse represented as follows:

$$\xi = \left\|\bar{\theta}\right\|^n + \|\bar{\omega}\|^n - 1, \tag{15}$$



(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 7. Simulation results of phase portraits for a desired orbit drawn as a spiric section. Red lines represent the phase portraits of the controllers, and the black line represents the desired trajectory. Blue points denote the results that the input is incomputable from the initial state.



(a) P-SW (b) P-LW (c) P-WO (d) VDDO (e) VDDO-IR

Fig. 8. Simulation results of orbit error in the case of a spiric section. Red lines indicate the results when the input u is always computable in the simulation period. Blue lines denote the results when the input u becomes incomputable in the simulation period, and the values in the graph are obtained from those initial conditions.

TABLE II

ROOT-MEAN-SQUARE ERROR OF SIMULATION AND EXPERIMENTAL RESULTS.

		Ellipse		Super-ellipse		Spiric section	
		e_{W}	e_{WO}	e_{W}	e_{WO}	e_{W}	e_{WO}
Sim.	P-SW	0.006	0.006	0.042	0.042	0.001	0.001
	P-LW	0.096	0.096	0.234	0.234	0.015	0.015
	P-WO	0.872	0.006	4.250	0.042	2.085	0.001
	VDDO	0.200	0.005	321.363	321.363	0.218	0.001
	VDDO-IR	2.155	2.203	35.518	39.425	4.109	3.959
Exp.	P-SW		0.054		0.124		0.017

where n is a constant and is more than 2 in this letter. In this case, the control input by (4) is represented as

$$u = -\frac{1}{\xi_\omega}\Big(\xi_\theta\omega + K(\boldsymbol{q})\big(\|\bar{\theta}\|^n + \|\bar{\omega}\|^n - 1\big)\Big), \tag{16}$$
$$\xi_\theta = n\alpha_\theta^n\|\theta\|^{n-1}\mathrm{sgn}(\theta),$$
$$\xi_\omega = n\alpha_\omega^n\|\omega\|^{n-1}\mathrm{sgn}(\omega),$$

where $\mathrm{sgn}(*)$ denotes the sign function. (16) indicates that the zero-division occurs independently of the gain design when $\omega = 0$ rad/s because $n > 2$.

We conducted the simulation under the condition that $K_\omega = 10.0\ \mathrm{s}^{-1}$ and $n = 4.0$. The resultant phase portraits are shown in Fig. 5, and the time variation in error is shown in Fig. 6. Similar to the case of the unit circle, trajectories that are computable in the 5-s simulation are represented by red lines in both figures. Blue and green lines represent the results that become incomputable from the initial state and the middle of the simulation, respectively. Table II presents the RMSEs of each controller in this case.

As shown in Figs. 5(d) and 6(d), several trajectories of VDDO diverge because of zero-division. This is because $\xi_\theta\omega$ cannot cancel ξ_ω, even when using the gain design (13). However, controllers, except for VDDO, show the red lines in Figs. 5 and 6. Therefore, regularization is effective to avoid zero-division.

Similar to the case of the unit circle, focusing on controllers except for VDDO, the VDDO-IR and P-WO cannot start from the initial state when $\xi_\omega = 0$. Due to this immobility, the difference between e_{W} and e_{WO} is observed in Table II. Comparing P-SW with P-LW, P-LW exhibits large errors due to the large weight. Therefore, the proposed controller with an initial input and small weight shows the best performance in the super-ellipse case.

D. Simulation result 3: Spiric section

Thirdly, we consider a spiric section to investigate the control performance if the desired orbit has two loops and cross points. The spiric section is illustrated by black lines in Fig. 7 and is represented as follows:

$$\xi = \left(\bar{\theta}^2 + \bar{\omega}^2\right)^2 - 2C_1\bar{\theta}^2 - 2C_2\bar{\omega}^2 + C_3, \tag{17}$$

where C_1, C_2, and C_3 are dimensionless constants. In this case, the control input by (4) is represented as

$$u = -\frac{1}{R_\omega}\Big(R_\theta\omega + K(\boldsymbol{q})R\Big), \tag{18}$$
$$R_\theta = 4\alpha_\theta\bar{\theta}\big(\bar{\theta}^2 + \bar{\omega}^2 - C_1\big),$$
$$R_\omega = 4\alpha_\omega\bar{\omega}\big(\bar{\theta}^2 + \bar{\omega}^2 - C_2\big).$$

Considering $\xi_\omega = 0$, (18) becomes incomputable when $\bar{\omega} = 0$ or $\bar{\theta}^2 + \bar{\omega}^2 - C_2 = 0$. Between these conditions, the former is avoidable if K is designed by (13); however, the latter remains.

We conducted the simulation with the condition $K_\omega = 100.0\ \mathrm{s}^{-1}$, $C_1 = 1.21$, $C_2 = 1.0$, and $C_3 = 1.0$. The resultant phase portraits are shown in Fig. 7, and the time variation in the trajectory error is shown in Fig. 8. Similar to the abovementioned cases, we represented the trajectories that are computable during the 5-s simulation by red lines in both figures. Blue lines or points represent the results that become incomputable from the initial state. Table II presents the RMSEs of each controllers in this case.

As illustrated in Fig. 7(d), VDDO becomes incomputable when $\bar{\theta}^2+\bar{\omega}^2-C_2 = 0$, as expected. Moreover, no trajectory is formed through the inner loop, even when VDDO is computable, because the resultant trajectories pass the outer of the desired orbit. This trend is observed in the result of P-LW. Meanwhile, P-WO and P-SW describe trajectories through the inner loop, indicating that the proposed controller with a small weight can pass near the entire points of the desired orbit.

Focusing on the error shown in Fig. 8 and Table II, VDDO exhibits the smallest error of e_{W}. However, as aforementioned, its result suffers from zero-division, and then e_{W} becomes large. Compared with the error except that of VDDO, P-SW exhibits totally small errors. Therefore, P-SW is the best controller for the spiric section.

V. Experimental results

A. Experimental setup

To investigate the performance of the proposed controller in a real robot, we conducted experiments using the 1-DOF robot presented in Fig. 1. As the joint axis corresponds to the vertical axis, the robot dynamics is represented as follows:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \omega \\ I^{-1}\big(\tau - b(\omega)\big) \end{bmatrix}, \tag{19}$$

where I is the joint inertia, τ is the joint torque, and $b(\omega)$ is the friction term. To make (19) close to (1), we designed τ as follows:

$$\tau = \hat{I}u + \hat{b}(\omega), \tag{20}$$

where $\hat{*}$ denote the identified value of $*$ through preliminary experiments. As friction remained slightly even when using the friction compensation $\hat{b}(\omega)$, we re-tuned the gain K_ω before each experiment to overcome the remaining friction. By trial and error through preliminary experiments, the gains for the unit circle, super-ellipse, and spiric section were set to $K_\omega = 50.0\ \mathrm{s}^{-1}$, $100.0\ \mathrm{s}^{-1}$, and $200.0\ \mathrm{s}^{-1}$, respectively. In the following experiments, we set the sampling interval ΔT to 2 ms.

B. Experimental results for each trajectory

First, we evaluated the control performance of P-SW for the same orbits in Section IV. In the experiments, the initial states were set to $\bar{\boldsymbol{q}}_0 = \boldsymbol{0}$, and the end time t_f was set to 10 s. The resultant phase portraits are shown in Fig. 9, and the time variations in the error are shown in Fig. 10. Additionally, the time variation in the normalized angle $\bar{\theta}$ is shown in Fig. 11. As the simulation results indicate that e_{W} and e_{WO} of P-SW are the same, we present only e_{WO} in Table II.

Unlike the simulation results, the resultant phase portraits shown in Fig. 9 are affected by the angular velocity noise. This noise is mainly attributed to the slight variation in sampling time. Because of this noise, e_{WO} of experiments are larger than those of simulations. Fig. 9 shows that the trajectory described by the proposed controller can start from $\bar{\boldsymbol{q}}_0 = \boldsymbol{0}$, similar to the above simulations. Moreover, the resultant portraits converged to approximately the neighborhood of each desired orbit. Furthermore, the result of the spiric section, shown in Figs 9(c) and 11(c), indicates that the resultant trajectory can go around the desired orbit with overlapping loops. Therefore, the proposed controller is available in a real robot.

C. Experimental result with human interaction

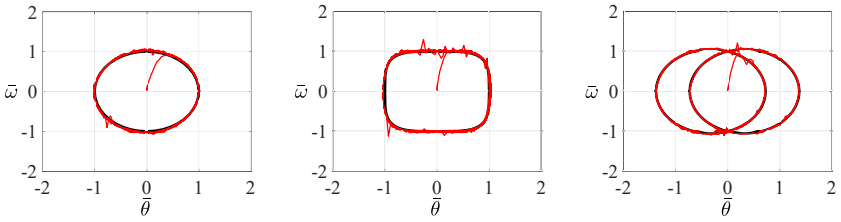
Finally, to investigate the restartable characteristic of P-SW, we conducted an experiment including physical interactions with a human. The desired orbit is the same spiric section as that discussed in Section IV. During the experiment, the human forcibly stopped the robot and released it repeatedly.

Experimental results are illustrated in Fig 12. As shown in Fig. 12, the human attempted to stop the robot four times: from 5.0 to 7.3 s, 12.4 to 13.6 s, 19.4 to 21.4 s, and 25.5 to 26.8 s. Fig.12(b) shows that the error increases during these periods. As shown in Fig. 12(d), the input values just before the robot stops are maintained until the human releases the robot. After those stopping periods, the robot restarts and converges to the desired orbit again. In particular, the first period is almost in a state that causes zero-division. Therefore, the proposed controller can restart from unfavorable states.

VI. CONCLUSIONS

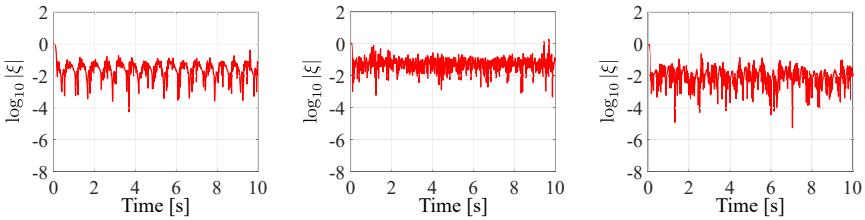
In this letter, we proposed a method to systematically design a time-independent and computable controller based on the virtual dynamics of the desired orbit. To ensure computability even when the computation includes zero-division, we applied the ZDA smoother to the computation of the control input. The performance of the proposed controller was investigated through simulations and experiments for a unit circle, super-ellipse, and spiric section. These results confirmed that the proposed controller could avoid zero-division while approaching the desired orbits. Furthermore, a physical interaction experiment showed that the proposed controller enabled the robot to restart from an unfavorable state and approach the desired orbits again.

In future studies, we will develop methods for obtaining the desired orbit. Notably, to use the proposed design, the desired orbit must be predefined as a formula. However, the curve fitting, e.g., the spline function or NURBS[14], will be used to relax this limitation. Additionally, the simulation results indicate that the gain value relates to the desired orbit. Therefore, gain tuning is another essential area of future work. Moreover, we will expand the proposed design to other



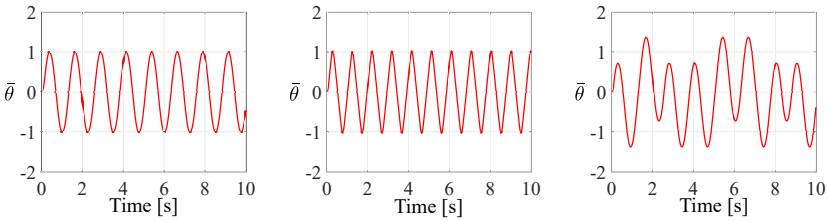
(a) Unit circle (b) Super-ellipse (c) Spiric section

Fig. 9. Experimental results of phase portraits. The red lines are phase portraits of the controllers, and the black line is the desired orbit.



(a) Unit circle (b) Super-ellipse (c) Spiric section

Fig. 10. Experimental results of the time variation of orbit error.



(a) Unit circle (b) Super-ellipse (c) Spiric section

Fig. 11. Experimental results of the time variation of the normalized angle $\bar{\theta}$.

spaces such as the configuration or three-dimensional spaces. In these spaces, unlike the phase space considered in this study, the flow direction should be defined in addition to the converging direction.

APPENDIX

A. Zero-division-avoidable smoother

The LPF is a well-known smoother of data such as measurement or control input. A 1st-order LPF is represented in the following discretized form:

$$y_k = \frac{w_F}{1+w_F} y_{k-1} + \frac{1}{1+w_F} x_k, \tag{21}$$

where y is the output from the LPF, and x is the input to the LPF. The subscript k denotes the variable at the time $k\Delta T$, and ΔT is the sampling interval. w_F denotes the weight that determines the internal division ratio between y_k and x_{k+1}. (21) can be derived from the following evaluation function:

$$E = \frac{1}{2}(x_k - y_k)^2 + \frac{1}{2} w_F (y_k - y_{k-1})^2. \tag{22}$$

Considering the stationary condition of (22) for y_k, $\frac{\partial E}{\partial y_k} = 0$, leads (21). Note that (23) is also known as the Tikhonov smoother [25].

Now, let us consider a case in which x_k is represented as $x_k = a_k^{-1} b_k$ using the variables a_k and b_k. If $a_k = 0$, zero-division occurs in the calculation of x_k, and thus, (21) will be incomputable. To address this issue, the author has previously proposed a filter that functions like a LPF even when zero-division occurs [5], [23]. This filter is based on the idea of (22) and is derived from the following evaluation function:

$$E = \frac{1}{2}(b_k - a_k y_k)^2 + \frac{1}{2} w_F (y_k - y_{k-1})^2. \tag{23}$$

Similar to the derivation of (21), the stationary condition of (23) for y_k leads to the following filter:

$$y_k = \frac{w_F}{a_k^2 + w_F} y_{k-1} + \frac{a_k}{a_k^2 + w_F} b_k. \tag{24}$$

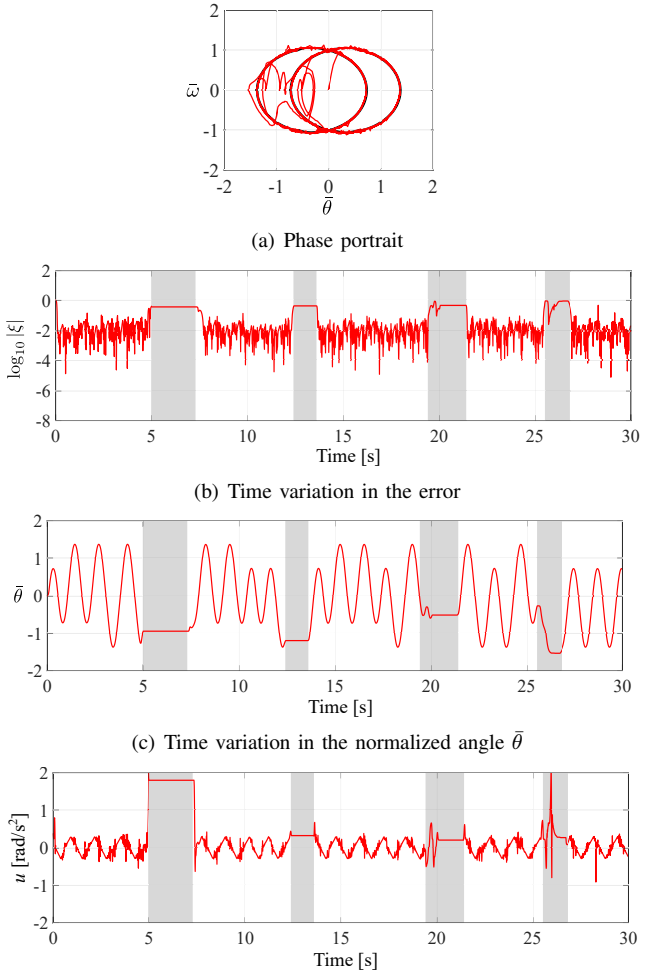
Obviously, y_k is equal to y_{k-1} when $a_k = 0$ in (24). Therefore, y_k is always computable, and we termed (24) the zero-division-avoidable (ZDA) smoother. If $a_k \neq 0$, (24) can be rewritten as follows:

$$y_k = \frac{w_a}{1+w_a} y_{k-1} + \frac{1}{1+w_a} x_k, \tag{25}$$

where $w_a = a_k^{-2} w_F$. (25) indicates that (24) is a 1st-order LPF with time-variable weight. Furthermore, the larger the value of $|a_k|$, the smaller the value of w_a, and the closer y_k is to $x_k = a_k^{-1} b_k$.

REFERENCES

[1] M. Okada, K. Tatani, Y. Nakamura, "Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion," *in Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation*, vol.2, Washington, DC, USA, 2002, pp.1410–1415.



(a) Phase portrait

(b) Time variation in the error

(c) Time variation in the normalized angle $\bar{\theta}$

(d) Time variation in the input u

Fig. 12. Experimental results with human interaction. Gray area means the period with physical interaction.

[2] M. Okada, K. Murakami, "Robot communication principal by motion synchronization using orbit attractor," *in Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, 2002, pp.2564–2569.

[3] K. Yamamoto and T. Shitaka, "Maximal Output Admissible Set for Limit Cycle Controller of Humanoid Robot," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, USA, 2015, pp.5690–5697.

[4] X. Tan, B. Zhang , G. Liu, X. Zhao, and Y. Zhao, A Time-Independent Control System for Natural Human Gait Assistance With a Soft Exoskeleton, *IEEE Transactions on Robotics*, vol.39, no.2, pp.1653–1667, 2023.

[5] K. Masuya, Y. Ide, M. Okada, "Assist-as-needed control based on region attractor for power-assist of knee joint taking into account differences in human and orthotic motion," *Transactions of the JSME (in Japanese)*, vol.91 no.941 , 24-00194, 2025.

[6] T. Sugihara, "Dynamics Morphing from Regulator to Oscillator on Bipedal Control," *in Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* St.Louis, 2009, pp.2940–2945.

[7] Y. Harada, J. Takahashi, D. Nenchev and D. Sato, "Limit Cycle Based Walk of a Powered 7DOF 3D Biped with Flat Feet," *in Proc. of the 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp.3623–3628.

[8] Y. Kim, Y. Tagawa, G. Obinata, K. Hase, "Robust control of CPG-based 3D neuromusculoskeletal walking model," *Biological Cybernetics*, vol.105, pp.269–282, 2011.

[9] T. G Sugar, A. Bates, M. Holgate, J. Kerestes, M. Mignolet, P. New, and R. K. Ramachandran, S. Redkar, and C. Wheeler, "Limit Cycles to Enhance Human Performance Based on Phase Oscillators," *Journal of Mechanisms and Robotics*, vol.7, no.1, 011001, 2015.

[10] G. Garofalo and C. Ott "Limit Cycle Control Using Energy Function Regulation with Friction Compensation," *IEEE Robotics and Automation Letters*, vol.1, no.1, pp.90–97, 2015.

[11] J. De La Fuente, S. C. Subramanian, T. G. Sugar, S. Redkar, "A robust phase oscillator design for wearable robotic systems," *Robotics and Autonomous Systems*, vol.128, 103514, 2020.

[12] R. Nasiri , M. Shushtari , H. Rouhani , and A. Arami, "Virtual Energy Regulator: A Time-Independent Solution for Control of Lower Limb Exoskeletons," *IEEE Robotics and Automation Letters*, vol.6, no.4, pp.7699–7705, 2021.

[13] R. Nasiri, H. Dinovitzer, N. Manohara, and A. Arami, "Coordinated human-exoskeleton locomotion emerges from regulating virtual energy," *PloS one*, vol.20, no.1, e0292334, 2025.

[14] M.-Y. Cheng and Y.-H. Wang, "Velocity Field Construction for Contour Following Tasks Represented in NURBS Form," *IEEE Transactions on Automatic Control*, vol.54, no.10, pp.2405–2410, 2009.

[15] A. Duschau-Wicke, J. von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener. "Path Control: A Method for Patient-Cooperative Robot-Aided Gait Rehabilitation," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol.18, no.1, pp.38–48, 2010.

[16] A. Martínez, B. Lawson , and M. Goldfarb, "A Controller for Guiding Leg Movement During Overground Walking With a Lower Limb Exoskeleton," IEEE Transactions on Robotics, vol.34, no.1, pp.183–193, 2018.

[17] H. J. Asl, T. Narikiyo, and M. Kawanishi, "An Assist-as-Needed Velocity Field Control Scheme for Rehabilitation Robots," *in Proc. of 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Madrid, Spain, 2018, pp.3322–3327

[18] A. Martínez, B. Lawson , C. Durrough, and M. Goldfarb, "A Velocity-Field-Based Controller for Assisting Leg Movement During Walking With a Bilateral Hip and Knee Lower Limb Exoskeleton," *IEEE Transactions on Robotics*, vol.35, no.2, pp.307–316, 2019.

[19] H. J. Asl, M. Yamashita , T. Narikiyo , and M. Kawanishi, "Field-Based Assist-as-Needed Control Schemes for Rehabilitation Robot," *IEEE/ASME Transactions on Mechatronics*, vol.25, no.4, pp.2100–2111, 2020.

[20] A. Shiriaev, J. W. Perram, and C. Canudas-de-Wit, "Constructive Tool for Orbital Stabilization of Underactuated Nonlinear Systems: Virtual Constraints Approach," *IEEE Transactions on Automatic Control*, vol.50, no.8, pp.1164–1176, 2005.

[21] R. D. Gregg, T. Lenzi, L. J. Hargrove, and J. W. Sensinger, "Virtual Constraint Control of a Powered Prosthetic Leg: From Simulation to Experiments With Transfemoral Amputees," *IEEE Transactions on Robotics*, vol.30, no.6, pp.1455–1471, 2014.

[22] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, vol.1, no.1, pp.1–16, 1972,

[23] K. Masuya and T. Sugihara, "Dead reckoning for biped robots that suffers less from foot contact condition based on anchoring pivot estimation," *Advanced Robotics*, vol.29, no.12, pp.785–799, 2015.

[24] O. Yuri, L. Acho, and L. Aguilar, "Quasihomogeneity approach to the pendubot stabilization around periodic orbits," *IFAC Proceedings Volumes*, vol.37, no.21, pp.411–416, 2004.

[25] A. Cortiella, K.-C. Park, and A. Doostan, "A Priori Denoising Strategies for Sparse Identification of Nonlinear Dynamical Systems: A Comparative Study," *Journal of Computing and Information Science in Engineering*, vol.23, no.1, 011004, 2023.