



Gait pattern generation algorithm for lower-extremity rehabilitation–exoskeleton robot considering wearer’s condition

Seung Hoon Hwang¹ · Dong Ik Sun¹ · Jeakweon Han¹ · Wan-Soo Kim¹

Received: 8 May 2020 / Accepted: 10 December 2020

© The Author(s), under exclusive licence to Springer-Verlag GmbH, DE part of Springer Nature 2021

Abstract

Various exoskeleton robots are being developed according to users’ conditions. Lower limb exoskeleton robots have attracted much attention for performing walking motions, a basic exercise for paralyzed patients. In this study, the walking environment of an exoskeleton robot wearer who walks using a crutch is identified as the foot position of the crutch support point and the support leg; therefore, the upcoming foothold location is calculated and sets as a control target. Rather than using preset pedestrian patterns, pedestrian patterns that can match the environment are created using the dynamic movement primitives machine learning technology. We evaluated the exoskeleton robot through an experiment with a healthy 29-year-old male. The experimental results showed that the exoskeleton robot was able to demonstrate the adaptive strides according to the wearer’s intention.

Keywords Rehabilitation robot · Exoskeleton robot · Gait pattern generation

1 Introduction

A person may be completely paralyzed by a strong external force applied to the spinal cord (e.g., in a car accident or a fall) resulting in damage to motor nerves, or caused by stroke, thereby completely losing the ability to walk and having limited potential for recovery [1]. There are 250,000–500,000 spinal cord injury (SCI) patients worldwide each year [2]. The priority for these patients is to restore their ability to walk, thus improve their quality of life [3]. Toward this end, many exoskeleton robots are being developed to enable patients to recover their walking ability. Commercial exoskeletal robots, such as ReWalk Robotics’ Rewalk [4], Ekso Bionics’ Ekso [5], and Parker Hannifin Corp’s Indego [6], enable the patient to walk with the help of crutches for safety. Patients use different types of support points for support each time they wear the robot and walk.

However, because exoskeleton robots move using a general person’s average walking pattern that is preset without considering the foot’s support position and the crutches,

the patient must walk according to the walking pattern. To improve the exoskeleton motion control, the exoskeleton robot needs to consider certain important aspects. First, when the patient uses the crutches, it is necessary to understand the walking environment, considering the crutches supporting position and the patient’s posture. Second, by creating a walking pattern tailored to the walking environment, the robot can walk accordingly.

Recently, various global institutes research the exoskeleton robot system motion control for SCI patients in various ways [7–9]. In one of the ways, research institutes have used several adaptive algorithms. Chen et al. used the joint angle and foot reaction force to determine the walking conditions and design an adaptive control strategy using a multilayer perceptron neural network algorithm [10]. Koller et al. used an electromyography sensor to determine the walking conditions and make the adaptive gain proportional to the myoelectric controller, which reduces the hip power during walking [11]. In some cases, adaptive methods have been applied to end-effector type gait robots, and studies have been conducted to increase the rehabilitation effect using adaptive algorithms for overground gait trainer and hand rehabilitation robots through wires [12–15]. As described above, studies to increase the rehabilitation effect by adaptive methods to rehabilitation robots are being conducted in various directions [16, 17].

✉ Jeakweon Han
jkhan@hanyang.ac.kr

Seung Hoon Hwang
hshhln@hanyang.ac.kr

¹ Hanyang University, Ansan Campus, Ansan-si,
Gyeonggi-do, Republic of Korea

The specific contributions of this paper are: (1) obtaining the feasible foothold position, which is calculated by the kinematic model of the exoskeleton as well as the crutches, that takes into account to keep the support polygon as well as patient's safety, (2) development of the gait pattern generation algorithm using the dynamic movement primitives (DMP) machine learning method that enables to generate the online adaptive walking pattern as regarding the environments. This approach exploits the developed exoskeleton, which is called Walking Assistant robot for Paraplegia Ver. 2 (WAP-2), and an inertial measurement unit (IMU) sensor, where is mounted in the upper limb of WAP-2 and the crutches, respectively.

The rest of this paper is organized as follows. Section 2 presents the support point identification method and the calculation of the support polygon loss target position using the kinematic model of the WAP-2 robot. Section 3 details the DMP technique and the creation of walking patterns. Section 4 presents simulations and an experiment to validate the research. In Sect. 5, we discuss our results. Finally, Sect. 6 presents the conclusions of this study.

2 WAP-2 modeling and support point detection

2.1 WAP-2 lower-limb exoskeleton robot

WAP-2 is a robot developed for restoring the walking ability of patients with paraplegia caused by SCI. Figure 1 shows the WAP-2 robot when worn. WAP-2 has eight joints, four active and two passive, in the sagittal plane. The four active joints support the flexion/extension movements of the hip and knee joints, and the two passive joints have the degree of freedom (DOF) of the plantar-/

dorsi-flexion motion of the ankle. Misalignment occurs because gait is not merely characterized by a rotational motion of 1 DOF of each joint but of a motion in three dimensions. Since this causes the wearer discomfort, in this study, WAP2 has two passive joints in the frontal plane that have the degree of freedom of the abduction/adduction motion of the hip joint (Fig. 2). The angle change of the active and passive joints is measured by the encoder, and the upper body's inclination is measured by the roll/pitch/yaw sensor and the angular velocity data of the upper back IMU sensor.

2.2 System kinematic modeling and supporting

Figure 3 shows the robot's product of exponential (POE) formula, which allows the modeling of its kinematics [18]. Before performing kinematic modeling, we made the following assumptions:

- No upper body movement occurs when the patient is being supported with crutches. In other words, there is no angle of the upper body in the supporting section of either crutch.
- Because the crutch length does not change, the crutch supporting point can be reliably obtained by calculating the configuration of both shoulders (Fig. 3, frames 7 and 8).

Hence, the forward kinematics have the form can be represented as

$$T_{06}(\theta) = e^{[S_0]\theta_0} e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_4]\theta_4} e^{[S_5]\theta_5} e^{[S_6]\theta_6} M_{06}, \quad (1)$$

Fig. 1 Walking assist robot for Paraplegia Ver. 2 (WAP-2)

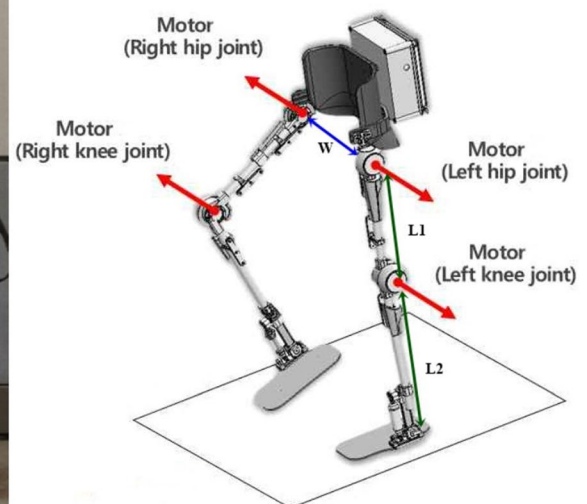


Fig. 2 WA-P hip module passive joint for human walking

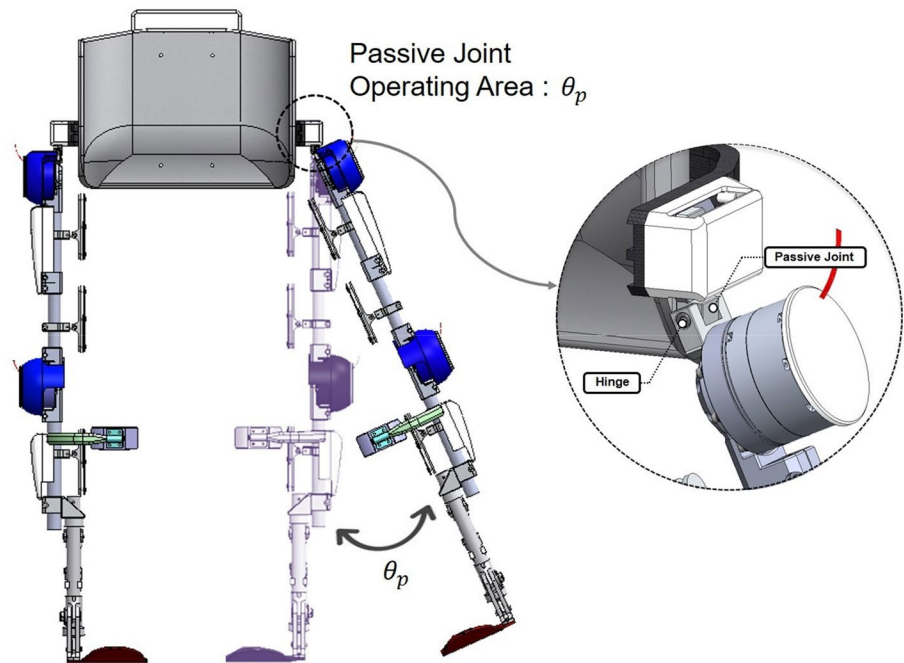
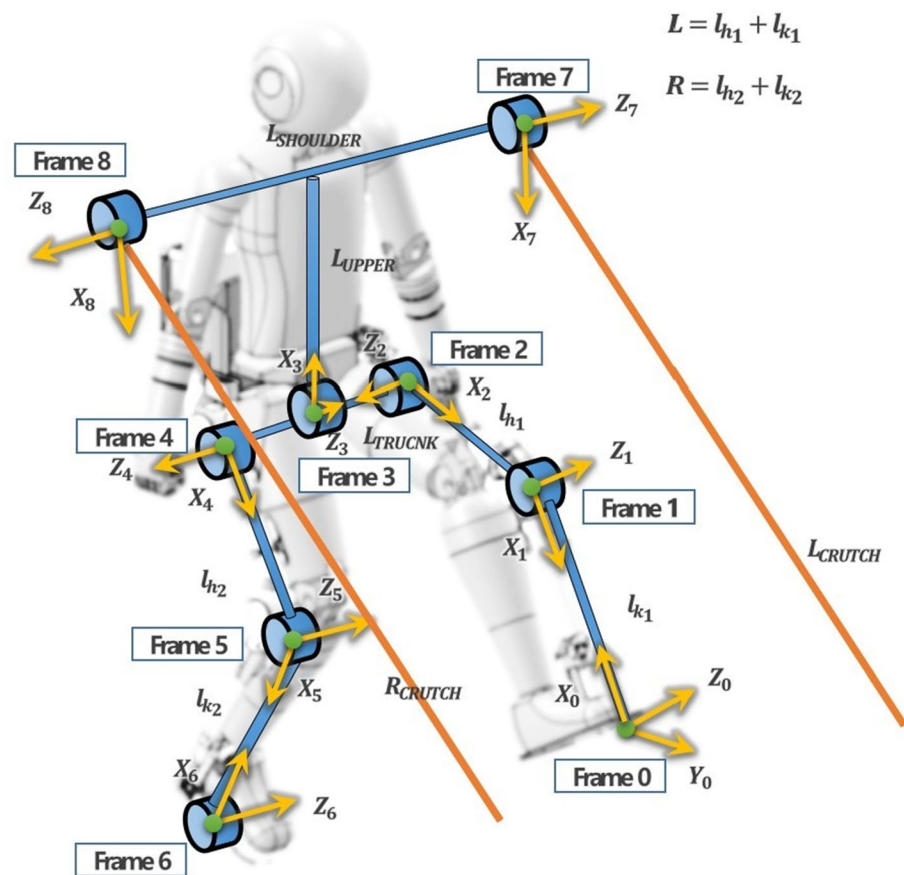


Fig. 3 PoE formula for WAP-2, (lk : length of robot calf link, lh : length of robot thigh link, L : left leg length $lk_1 + lh_1$, R : right leg length $lk_2 + lh_2$)



where T_{06} represents the frame 6 configuration relative to frame 0. To calculate the forward kinematics with POE formula, we need the following elements [18].

- The end-effector configuration $M \in SE(3)$ when the robot joints are at zero position.
- The screw axes $S_1, S_2, \dots, S_n$ expressed in the fixed frame correspond to the joint motions when the robot is at the joint zero position.
- The joint variables are $\theta_1, \dots, \theta_n$

Then, M_{06} is expressed as

$$M_{06}(\theta) = \begin{bmatrix} 1 & 0 & 0 & L-R \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{\text{trunk}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Table 1 lists the screw axes for frames 0 to 6. $S_i = [\omega_i^T v_i^T]^T$ is skew symmetric matrix. Similarly, at the base frame 0, for both shoulder frames 7 and 8 (Fig. 3), the forward kinematics are, respectively, expressed as follows:

$$T_{07}(\theta) = e^{[S_0]\theta_0} e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_7]\theta_7} M_{07}, \quad (3)$$

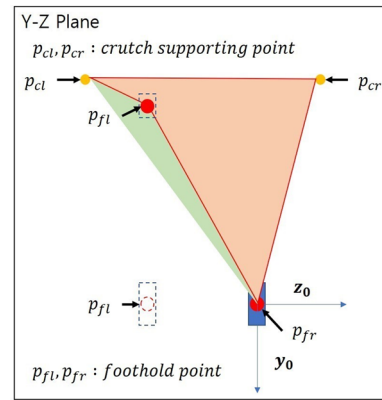
$$T_{08}(\theta) = e^{[S_0]\theta_0} e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_8]\theta_8} M_{08}, \quad (4)$$

$$M_{07}(\theta) = \begin{bmatrix} 1 & 0 & 0 & L + L_{\text{UPPER}} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -\frac{1}{2}L_{\text{trunk}} + 0.5L_{\text{shoulder}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

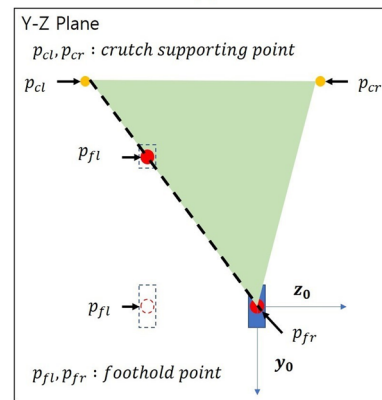
and

$$M_{08}(\theta) = \begin{bmatrix} -1 & 0 & 0 & L + L_{\text{UPPER}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -\frac{1}{2}L_{\text{trunk}} - 0.5L_{\text{shoulder}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

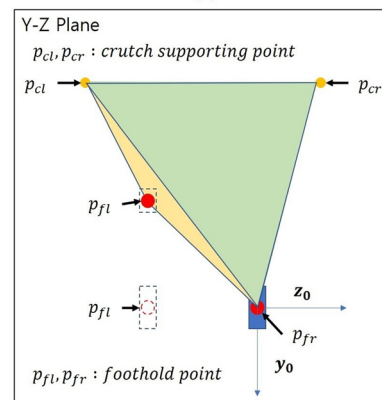
Both crutch supporting points p_{cr} and p_{cl} can be obtained by the 7 and 8 configuration because the length of the crutches is constant. If the x -axis component of the



(a)



(b)



(c)

Fig. 4 Supporting point of crutch and foot: **a** long stride length, **b** swing leg foothold on line equation, and **c** short stride length

Table 1 Screw axes of frames 0 to 6

i	ω_i	v_i
0	(0, 0, 1)	(0, 0, 0)
1	(0, 0, 1)	(0, -lk1, 0)
2	(0, 0, -1)	(0, -L, 1)
4	(0, 0, -1)	(0, -L, 0)
5	(0, 0, 1)	(0, L-lh2, 1)
6	(0, 0, 1)	(0, L-R, 0)

Table 2 Screw axes of frames 0 to 7

i	ω_i	v_i
0	(0, 0, 1)	(0, 0, 0)
1	(0, 0, 1)	(0, -lk1, 0)
2	(0, 0, -1)	(0, -L, 1)
3	(0, 0, 1)	(0, L, 0)
7	(0, 0, 1)	(0, L + L _{UPPER} , 0)

Table 3 Screw axes of frames 0 to 8

i	ω_i	v_i
0	(0, 0, 1)	(0, 0, 0)
1	(0, 0, 1)	(0, -lk1, 0)
2	(0, 0, -1)	(0, -L, 1)
3	(0, 0, 1)	(0, L, 0)
8	(0, 0, -1)	(0, -L -LUPPER, 0)

supporting point has a value of 0 according to the coordinates of the fixed frame 0, the foothold and the crutch supporting point of each foot in the z and y two-dimensional plane, shown in Fig. 4, indicate a location (Tables 2, 3).

The state before the swing is a quadruple stance (two feet, two crutches). However, the state when walking is a triple stance (one foot, two crutches), and the support polygon becomes a triangle, as shown in Fig. 4. If the swinging foot is inside the support polygon, it has the shape of a rectangular support polygon (Fig. 4a) that is smaller in area than the support triangle. In terms of stability, walking is more stable because the area of the support polygon can more easily enter the stable area than when it is reduced. However, if the stride length is too short (Fig. 4c), it slows down the walking speed and tires the patient's upper body after a long walk. If the stride length is too long, the area of the support polygon may decrease and cause the wearer's Center of pressure (COP) to move out of the support polygon, resulting in unstable walking. Therefore, we calculate the target foothold position for walking with two compromises. As shown in Fig. 4b, if the target position of the foothold of the swinging leg is made on the straight line, the equation of the triangle of the current support polygon can be expressed as

$$p_{fl_{target}} = \frac{(p_{fr_y} - p_{cl_y})}{(p_{fr_x} - p_{cl_x})} (p_{fl_{target}} - p_{fr_x}) + p_{fr_y} \quad (7)$$

In the linear equation derived by the foot position p_{fr} and the left crutch point p_{cl} we can calculate a target foot position for the swinging leg that does not reduce the support polygon space. Then, before swing foot position p_{cl} and target position in line equation make the target stride length $p_{fl_{target}}$. This can generate a walking pattern that does not reduce the support polygon space. Moreover, the target position changes according to how the patient places the crutch and positions the foot.

With this concept, the zero moment point (ZMP) generated from the two crutch points and one stance foot position is in the support polygon and it does not bring too short stride. It can thus be considered that it is possible to provide the wearer with safety and a stride adapted to his or her condition [19, 20].

3 Intuitive pattern generation

3.1 Dynamic movement primitives (DMPs)

DMPs apply biological movements to robotics technology to create appropriate patterns in different situations rather than using preset patterns. A DMP is considered a combination of simple policies aimed at reducing the control policy of complex movements such as walking. To that end, it generates a pattern of human motion as a sum of nonlinear dynamics. This method is very similar to the motor pattern generator method in neurobiology. The pattern can be generated without manual parameter tuning and therefore has the advantage of being able to cope with both rhythmic and discrete movements. In this study, the DMP theory was applied to the WAP-2 system to create walking patterns suitable for different situations [21].

3.2 Formalization of DMPs for WAP-2

DMPs were applied to WAP-2 by using the following second-order dynamic equation:

$$\ddot{z} = \alpha(\beta(g - y) - \dot{z}) + f, \dot{y} = z, \quad (8)$$

where y , $\dot{y}$, and $\ddot{y}$ are, respectively, the desired angular position, velocity, and acceleration of WAP-2, respectively, g is the known target state. α and β are the gain terms for the critically damped system. The forcing term nonlinear function f can be chosen as

$$\sum_{i=1}^N \Psi_i(t) \omega_i / \sum_{i=1}^N \Psi_i(t), \quad (9)$$

where Ψ_i represents fixed basis functions, and ω_i represents adjustable weights that depend on the forcing function's time [22].

The canonical dynamical system of our robot system is given by

$$\dot{x}(t) = -\alpha_x x(t), \quad (10)$$

where α_x is a constant that makes the system stable. By substituting time in (10), the forcing function can be expressed as

$$f(t) = \frac{\sum_{i=1}^N \Psi_i(t) \omega_i}{\sum_{i=1}^N \Psi_i(t)} x * (g - y_0), \quad (11)$$

where y_0 is the initial state of the system, and $\Psi_i(x)$ is a Gaussian function expressed as

$$\Psi_i(x) = \exp(-h_i(x - c_i)^2), \quad (12)$$

where ci is the center of the Gaussian function, and hi is the variance. Interestingly, the complex nonlinear function is represented by a linear combination of a simple basis function, which is a common method in machine learning [22]. By properly learning the ω_i in (11), reproduction can be performed according to the primitive trajectory.

3.3 Learning using locally weighted regression (LWR)

As previously stated, we aim to solve a complex modeling problem using many simple local models. To learn the ω_i , we use the locally weighted regression (LWR) method. LWR is a very fast learning process that is independent of kernels. This enables us to achieve stable parameterization for movement recognition. The weights ω_i define the spatial path with different weights of the basis function. Before calculating the ω_i , the desired trajectory of the f value can be expressed using (8) as

$$f_{\text{target}} = \dot{z} - \alpha(\beta(g - y) - z) \quad (13)$$

Then, we determine how to force function f into f_{target} . The LWR finds each basis function Ψ_i in the forcing function f of the corresponding ω_i that minimizes the locally weighted quadratic error as

$$J_i = \sum_{t=1}^N \Psi_i(t) (f_{\text{target}}(t) - \omega_i(x(t)(g - y_0)))^2 \quad (14)$$

The solution of (14) is

$$\omega_i = \frac{S^T \Gamma_i f_{\text{target}}}{S^T} \quad (15)$$

where

$$S = \begin{pmatrix} x_{t_0}(g - y_0) \\ x_{t_0}(g - y_0) \\ \vdots \\ x_{t_0}(g - y_0) \end{pmatrix}, \Gamma_i = \begin{pmatrix} \Psi_i(1) & 0 & \dots & 0 \\ 0 & \Psi_i(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Psi_i(N) \end{pmatrix},$$

$$f_{\text{target}} = \begin{pmatrix} f_{\text{target}}(1) \\ f_{\text{target}}(2) \\ \vdots \\ f_{\text{target}}(N) \end{pmatrix}$$

3.4 Rescaling learned walking pattern

To make variable walking trajectories, another term, k , is added in system Eqs. (3) and (5):

$$\dot{z} = k^2(\alpha(\beta(g - y) - z) + f), \quad \text{and} \quad (16)$$

$$\dot{x}(t) = k(-\alpha_x x(t)) \quad (17)$$

where the range of k is always the positive value and varies with the length of thigh and calf links that is defined the stride length. For example, for $k=0.8$, the new trajectory of the walking pattern has smaller stride length than that of the original walking pattern, whereas for $k=1.2$, the new trajectory of the walking pattern has longer stride length than that of the original walking pattern without losing the original walking pattern. Figures 5 and 6 show the simulation results obtained using the DMP method. For $k=1$, it can be confirmed that the reproduction trajectory is similar to the primitive trajectory and the stride length is 0.6 m, similar to the demonstration, as shown in Fig. 5. For $k>1$, the amplitude and stride length appear to be greater than those of the primitive trajectory, as shown in, as shown in Fig. 6. Through this generalization, the walking pattern can be regenerated so that the user can walk with a desired step length. This should result in increased utilization in activities of daily living (ADLs).

As mentioned in Sect. 2, we aim to produce an exoskeleton gait trajectory that considers the wearer's state. To find a walking stride that fits the wearer's state, we calculate the relationship between the stride length and k using curve fitting method.

The relationship between the desired stride length pfl_{target} and constant k is expressed as.

$$pfl_{\text{target}} = -0.994k^3 + 2.644k^2 - 1.37k + 0.2702 \quad (18)$$

When applied to WAP-2, this pattern generation method works as follows:

- The WAP-2 robot detects the wearer's state using the kinematics solutions of (1)–(6).
- Then, the robot calculates the appropriate desired stride length pfl_{target} .
- The robot can calculate a positive constant k using (18).
- Finally, the robot generates the rescaled gait trajectory using the DMP solution with a constant k .

To evaluate this pattern generation algorithm, we conducted an experiment, described in Sect. 4.

4 Experiment and evaluation

4.1 Experimental setup

The WAP-2 robot used in the experiment to evaluate out pattern generation method consists of an NI sbRIO-9627 main board and four Kollmorgen RBE-01510B actuators and is coded using NI LabView 2015. The crutches have an IMU sensor (Xsens MTi 1-series) for crutches pose and the back of WAP-2 have same an IMU for upper limb

Fig. 5 Simulation of DMP method for walking pattern generation ($k=0.8$). (Top) Two graphs show the trajectory of the hip and knee joints. Red and blue lines, respectively, indicate original walking pattern and rescaled walking pattern obtained using DMP method. (Bottom) Two graphs show the foot trajectory in the Cartesian space. Left: rescaled foot trajectory. Right: original foot trajectory. **a** During simulation, red link: thigh link 0.4 m, Blue link: Calf link 0.42 m and **b** simulation result

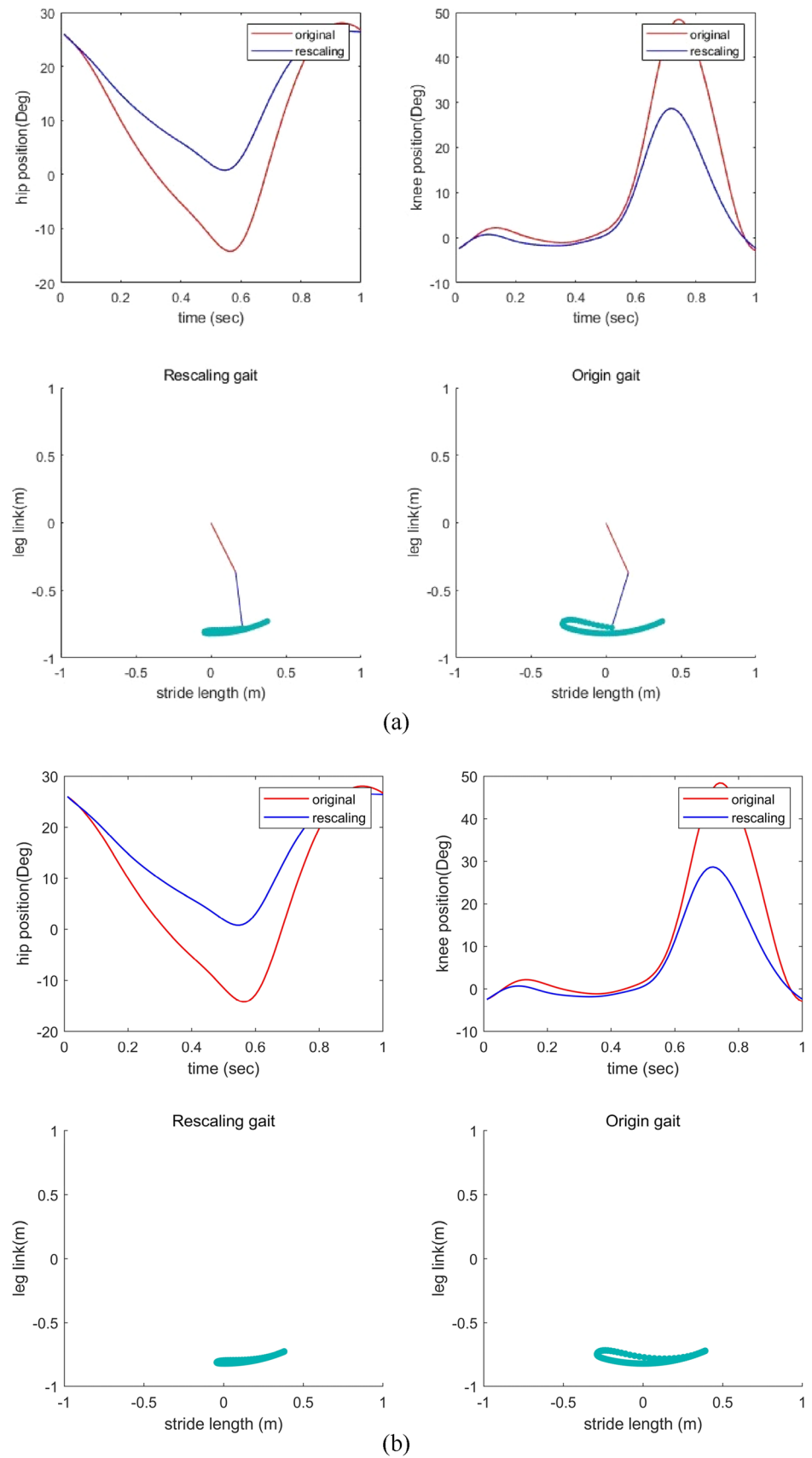
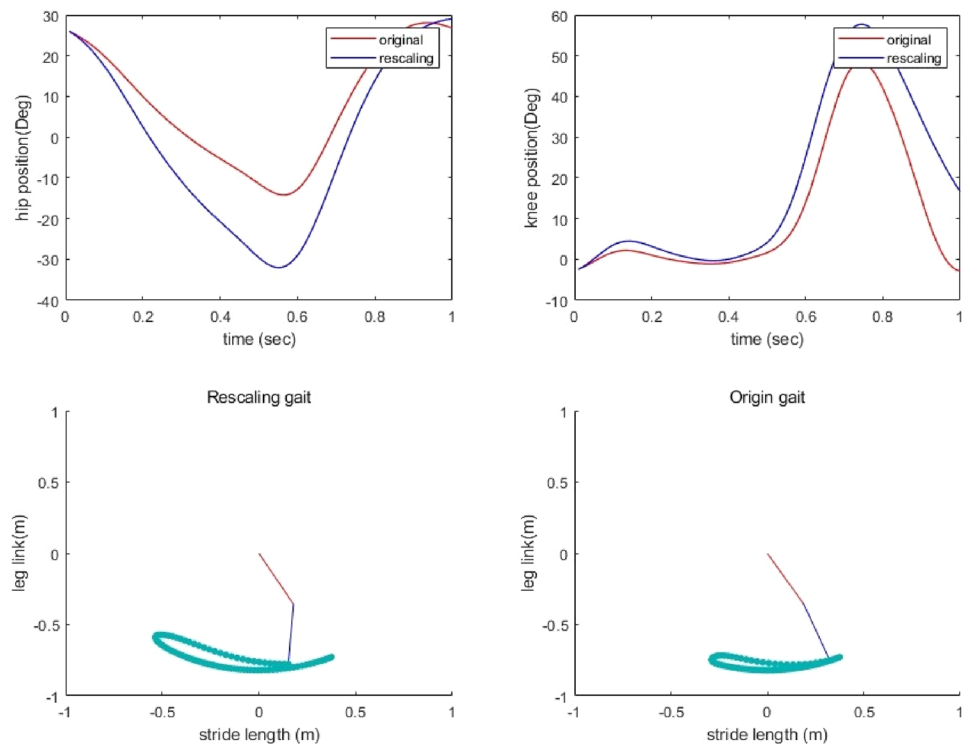
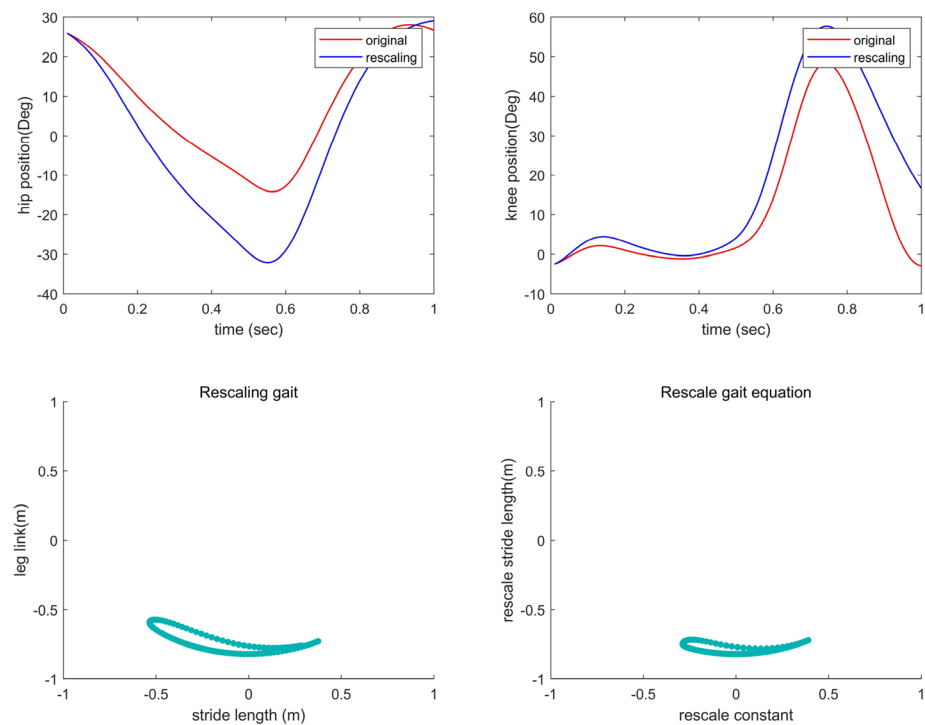


Fig. 6 Simulation of DMP method for walking pattern generation ($k = 1.2$). (Top) Two graphs show the trajectory of the hip and knee joints. Red and blue lines, respectively, indicate original walking pattern and rescaled walking pattern obtained using DMP method. (Bottom) Two graphs show the foot trajectory in the Cartesian space. Left: rescaled foot trajectory. Right: original foot trajectory. **a** During simulation, Red link: thigh link 0.4 m, Blue link: Calf link 0.42 m and **b** simulation result



(a)



(b)

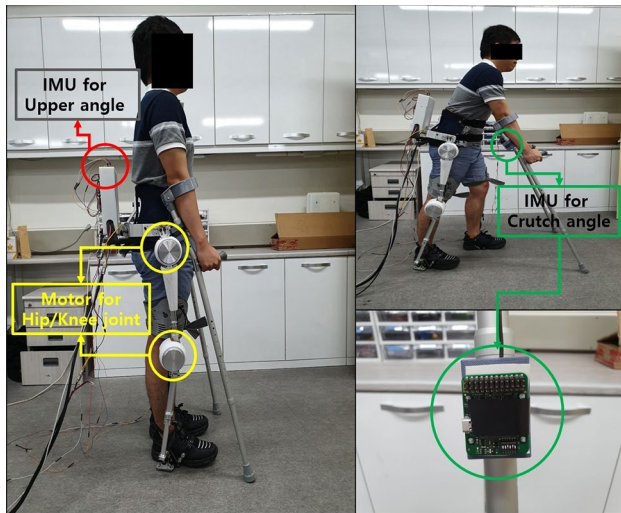


Fig. 7 Configuration of the experimental setup environment

lean angle [23]. The robot uses a proportional–differential (PD) control trajectory tracking method to implement the walking motion. The experiment was conducted with the participation of a healthy 29-year-old man wearing the WAP-2 robot (Fig. 7). For the validation experiment, the subject was asked to perform three steps for a trial. In each trial, the length of the step varied slightly from the holding position of crutches: long length 0.8125 m ($k = 1.4595$), normal 0.6298 m ($k = 1.0889$), and short 0.58 m ($k = 1.0322$).

4.2 Experimental result and evaluation

The experimental results of pattern generation are shown in Fig. 8. Figure 8a shows that WAP-2 calculates p_{fl_target} by combining the angles of each joint, the inclination of the upper body, and the angle of the crutch. The resulting p_{fl_target} is 0.7292 m. Since it has a larger value than the original stride length when $k = 1$, the new constant k is calculated $k = 1.22$, which is greater than original. The DMP creates a new joint trajectory. The blue line in Fig. 8a shows the robot's joint angle when it moves to the new trajectory. As can be seen, the amplitude of the joint angle is greater than the original.

Conversely, in Fig. 8b, WAP-2 calculates the p_{fl_target} from joint angle and IMU data. The new p_{fl_target} is 0.5027 m. Since this is less than the stride length when $k = 1$, the constant k also has a value less than 1 ($k = 0.95$). A new trajectory is generated by the DMP. The blue line in Fig. 8b shows the joint angle of the WAP-2 robot determined by the newly generated value. As can be seen, the amplitude of the joint angle is smaller than the original. The experimental results

show that the WAP-2 exoskeleton robot determined the wearer's state by the angle of each joint, the inclination of the upper body, and the inclination of the crutches. It also calculated a stride length that did not reduce the area of the support polygon and created an appropriate joint trajectory to provide movement to the wearer. We can thus see that the exoskeleton robot generates a walking pattern suitable for the wearer by generating either short or long strides according to the wearer's state. These results demonstrate that our proposed walking pattern generation method is valid.

5 Discussion

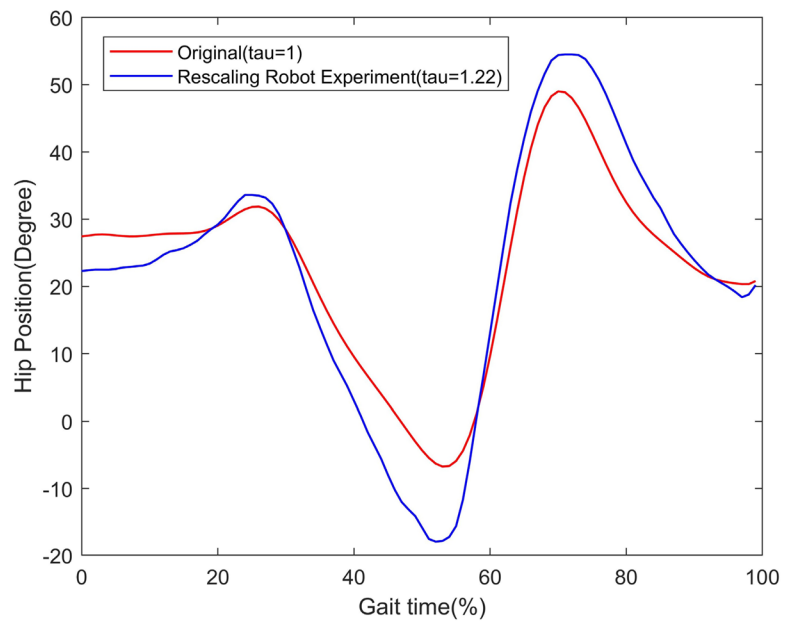
Through the proposed algorithm, the position of the crutch point and the stance foot were determined by using the wearer's kinematic information. A new hip and knee joint trajectory were, respectively, generated using the DMP algorithm. Our simulation results showed that when the constant k was greater than 1, it produced a longer stride than the original, and when it was less than 1, it resulted in a shorter stride than the original. This aspect could also be seen in the hip and knee joint trajectories. The results of our experiment, which was demonstrated by a real exoskeleton robot, showed that the stride length and hip/knee joint patterns are changed by the wearer's kinematic state. The main advantage of the proposed method is in its reduced complexity and limited amount of required measurement systems, which could significantly improve its applicability in real environments.

The objective of the proposed method is to exploit the potential of the exoskeleton gait control using in the rehabilitation field. However, it should be emphasized that this result was produced with a healthy participant. Accordingly, future works will focus on the investigation of the subjects' experiment to the paraplegia patients and its feasibility that includes the long-term effects on the subjects as well as the comparative study (i.e., comparison with the traditional approach). Also, the capture point, the zero rate of angular momentum, and the center of pressure considered when creating a humanoid gait pattern are additionally used, the gait pattern generation algorithm proposed in this study is expected to be further developed [24, 25].

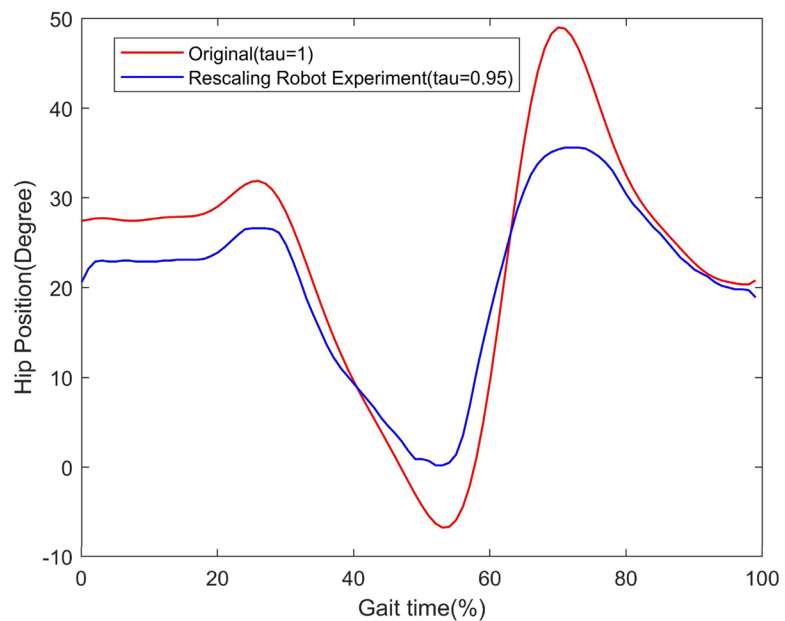
6 Conclusion

This study proposed a method for generating various and appropriate for an exoskeleton WAP-2 robot. The trajectories were learned and reproduced by DMP. Although many exoskeletons for SCI patients have been studied, the previous work has employed mostly to generate normal gait pattern to users without taking into account the patient's state and

Fig. 8 Measuring the stride length and joint angle of each k scale. **a** Result trajectory when $k = 1.22$, **b** result trajectory when $k = 0.95$



(a) The Hip joint experiment WAP-2 data red-line : The original(when $k=1$) hip joint data, blue-line : The rescaling($k=1.22$) hip joint data when $p_{f l_{y target}} = 0.7292\text{m}$ longer than original stride length



(b) The Hip joint experiment WAP-2 data red-line : The original(when $k=1$) hip joint data, blue-line : The rescaling($k=0.95$) hip joint data when $p_{f l_{y target}} = 0.5027\text{m}$ less than original stride length

its effectiveness on varying patterns. We evaluated this gait pattern method through simulation and an experiment using a real exoskeleton robot with a healthy wearer. Our results represented the potential to the prior approach that it can

be used in the further applications such as the daily life-style changes through the safety of the patient's exoskeleton walking.

This method can be useful in ADLs that the various stride lengths can help to avoid obstacles, such as sidewalk curbs or potholes, and prevent falling forward by increasing the stride length. Moreover, the safe supporting point's considerations can be employed a stroke patient's safety while walking with the exoskeleton and crutches. However, it is necessary to further our research with experiments involving actual SCI or stroke patients.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1F1A1061246).

References

1. American College of Surgeons (2008) Committee on Trauma. ATLS, advanced trauma life support for doctors: student course manual. American college of surgeons
2. World Health Organization and International Spinal Cord Society (2013) International perspectives on spinal cord injury. World Health Organization, Geneva
3. Simpson LA, Eng JJ, Hsieh JTC, Wolfe, and Dalton L the Spinal Cord Injury Rehabilitation Evidence (SCIRE) Research Team (2012) The health and life priorities of individuals with spinal cord injury: a systematic review. *J Neurotrauma* 29(8):1548–1555
4. Esquenazi A, Talaty M, Packel A, Saulino M (2012) The rewalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *Am J Phys Med Rehabil* 91(11):911–921
5. Bionics E (2015) Ekso. *Recuperado* 30(07):2015
6. Lajeunesse V, Vincent C, Routhier F, Careau E, Michaud F (2016) Exoskeletons' design and usefulness evidence according to a systematic review of lower limb exoskeletons used for functional mobility by people with spinal cord injury. *Disabil Rehabil Assist Technol* 11(7):535–547
7. Esquenazi A, Talaty M, Jayaraman A (2017) Powered exoskeletons for walking assistance in persons with central nervous system injuries: a narrative review. *PM&R* 9(1):46–62
8. Han Y, Zhu S, Zhou Y, Gao H (2019) An admittance controller based on assistive torque estimation for a rehabilitation leg exoskeleton. *Intel Serv Robot* 12(4):381–391
9. Kalani H, Akbarzadeh A, Nabavi SN, Moghimi S (2018) Dynamic modeling and CPG-based trajectory generation for a masticatory rehabilitation robot. *Intell Serv Robot* 11(2):187–205
10. Chen C, Liu D, Wang X, Wang C, Wu X (2017) An adaptive gait learning strategy for lower limb exoskeleton robot. In: 2017 IEEE international conference on real-time computing and robotics (RCAR). IEEE, pp 172–177
11. Koller JR, Jacobs DA, Ferris DP, Remy CD (2015) Learning to walk with an adaptive gain proportional myoelectric controller for a robotic ankle exoskeleton. *J Neuroeng Rehabil* 12(1):1–14
12. Tomelleri C, Waldner A, Werner C, Hesse S (2011) Adaptive locomotor training on an end-effector gait robot: evaluation of the ground reaction forces in different training conditions. In: 2011 IEEE international conference on rehabilitation robotics. IEEE, pp 1–5
13. Everarts C, Vallery H, Bolliger M, Ronsse R (2013) Adaptive position anticipation in a support robot for overground gait training enhances transparency. In: 2013 IEEE 13th international conference on rehabilitation robotics (ICORR). IEEE, pp 1–6
14. Hwang SH, Lee SC, Shin DB, Baek IH, Kim MJ, Sun DI, Kim BS, Hwang SW, Han CS (2019) Intuitive gait pattern generation for an exoskeleton robot. *Int J Precis Eng Manuf* 20(11):1905–1913
15. Chowdhury A, Nishad SS, Meena YK, Dutta A, Prasad G (2018) Hand-exoskeleton assisted progressive neurorehabilitation using impedance adaptation based challenge level adjustment method. *IEEE Trans Haptics* 12(2):128–140
16. Cafolla D (2019) A 3D visual tracking method for rehabilitation path planning. In: *New trends in medical and service robotics*. Springer, Berlin, pp 264–272
17. Cafolla D, Chen I-M, Ceccarelli M (2015) An experimental characterization of human torso motion. *Front Mech Eng* 10(4):311–325
18. Lynch KM, Park FC (2017) *Modern robotics*. Cambridge University Press, Cambridge
19. Di P, Huang J, Nakagawa S, Sekiyama K, Fukuda T (2013) Fall detection and prevention in the elderly based on the ZMP stability control. In: 2013 IEEE workshop on advanced robotics and its social impacts. IEEE, pp 82–87
20. Scianca N, De Simone D, Lanari L, Oriolo G (2020) MPC for humanoid gait generation: stability and feasibility. *IEEE Trans Robot* 36(4):1171–1188
21. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S (2013) Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput* 25(2):328–373
22. Bishop CM (2006) *Pattern recognition and machine learning*. Springer, Berlin
23. Cho YK, Kim K, Ma S, Ueda J (2018) A robotic wearable exoskeleton for construction worker's safety and health. In: *ASCE construction research congress*, pp 19–28
24. Pratt J, Carff J, Drakunov S, Goswami A (2006) Capture point: a step toward humanoid push recovery. In: 2006 6th IEEE-RAS international conference on humanoid robots. IEEE, pp 200–207
25. Al-Shuka HFN, Corves B, Zhu W-H, Vanderborght B (2016) Multi-level control of zero-moment point-based humanoid biped robots: a review. *Robotica* 34(11):2440

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.