EMG-Based Control of a Lower-Limb Power-Assist Robot

Kazuo Kiguchi[1] and Yoshiaki Hayashi[2]

[1] Kyushu University, Japan
[2] Saga University, Japan

Abstract. Power-assist robots are expected to work in many fields such as industry, military, medicine, etc. A lower-limb power-assist robot for physically weak persons is supposed to be used for self-rehabilitation or daily motion assist. In order to assist daily motion of the physically weak persons, the robot must estimate the motion intention of the user in real-time. Although there are several kinds of method to estimate the motion intention of the user in real-time, Electromyogram (EMG) signals are often used to estimate that since they reflect the users muscle activities. However, EMG-based real-time motion estimation is not very easy because of several reasons. In this chapter, an EMG-based control method is introduced to control the power-assist lower-limb exoskeleton robot in accordance with users motion intention. A neuro-fuzzy modifier is applied to deal with those problems. The problems of EMG-based motion estimation are cleared by applying the neuro-fuzzy modifier.

Sometimes there is a problem in the users motion even though the users motion is assisted, if the user misunderstands interaction between the users motion and a surrounding environment. In that case, the users motion should be modified to avoid an accident. In this chapter, a method of perception-assist is also introduced to automatically modify the users motion properly.

Keywords: EMG signal, Motion estimation, Intention estimation.

1 Introduction

Many studies on power-assist robots have been carried out up to the present since they are expected to work in many fields such as industry, military, medicine, etc.[1]-[15]. A lower-limb power-assist robot is especially important for physically weak persons to make daily living motion or to perform self-rehabilitation. In order to assist daily motion of the physically weak persons, the robot must estimate the motion intention of the user and assist the estimated users motion in real-time. Although there are several kinds of method to estimate the motion intention of the user in real-time [9]-[11], Electromyogram (EMG) signals are often used to estimate that since they reflect the users muscle activities. The magnitude of the EMG signal is almost proportional to the activity level of the muscle. However, EMG-based real-time motion estimation is not very easy to

S. Mohammed et al. (eds.), *Intelligent Assistive Robots,*
Springer Tracts in Advanced Robotics 106, DOI: 10.1007/978-3-319-12922-8_14

be realized for multi-DOF power-assist exoskeletons because 1: obtaining the same EMG signals for the same motion is difficult even with the same person, 2: activity level of each muscle and the way of using each muscle for a certain motion is different between persons, 3: one muscle is not only concerned with one motion but also another kinds of motion, 4: activity of antagonist muscles affects the generated joint torque, 5: the role of each muscle for a certain motion varies in accordance with joint angles (i.e., the posture change affects the role of each muscle for a certain motion), 6: the activity level of some muscles such as bi-articular muscles are affected by the motion of the other joint, and 7: the effect of antagonist muscle must be taken into account. Therefore, these problems must be cleared in order to apply an EMG-based control method to activate the power-assist robot. In this chapter, an EMG-based control method that can deal with above mentioned problems is introduced to control the power-assist lower-limb exoskeleton robot in accordance with users motion intention.

In the case of the power-assist robots which assist daily motion of physically weak persons, it is important to assist the users motion automatically according to the users intention. However, sometimes there is a problem in the users motion even though the users motion is assisted, if the user misunderstands interaction between the users motion and a surrounding environment. Since not only the motor ability but also the sensory ability is deteriorated in the case of some physically weak persons, the user tends to misunderstand interaction between the users motion and a surrounding environment. In that case, even though the user expands the motor activity with the power-assist robot, the user might not be able to perceive the risks associated with users motions correctly and be involved in an unexpected accident such as tumbling or falling. Therefore, the users motion should be modified to avoid the unexpected accident. In this chapter, a method of perception-assist is also introduced to automatically modify the users motion properly [18]. In the perception-assist, the robot monitors the interaction between the user and the environment using some sensors, and tries to modify the users motion automatically by adding additional force only if certain problems are found in the users motion. Furthermore, ZMP (Zero Moment Point) should be taken into account in order to make the stable lower-limb motion.

2 EMG-Based Motion Estimation

2.1 EMG

Skin surface electromyogram (EMG) is an electric signal generated when the muscle is activated. Since the EMG signals directory reflect the muscle activity levels, they are used as main input signals to estimate the generating lower-limb motion and control the lower-limb exoskeleton in accordance with the users motion intention. Since raw EMG signals are not suitable as input signals to the controller, feature of the raw signal must be extracted. There are many kinds of feature extraction methods, e.g., Root Mean Square, Mean Absolute Value, Mean Absolute Value Slope, Zero Crossings, Slope Sign Changes, or Waveform

Length [19]. One of the most effective feature extraction methods for the real-time human motion estimation is the root mean square (RMS). The RMS of EMG signal is calculated with the equation written below:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} v_i^2} \tag{1}$$

where N is the number of the segments ($N = 400$), v_i is the voltage at ith sampling. Figure 1 shows how the feature of the row EMG signal is extracted with the RMS calculation. The amount of the RMS shows the activity level of the muscle.

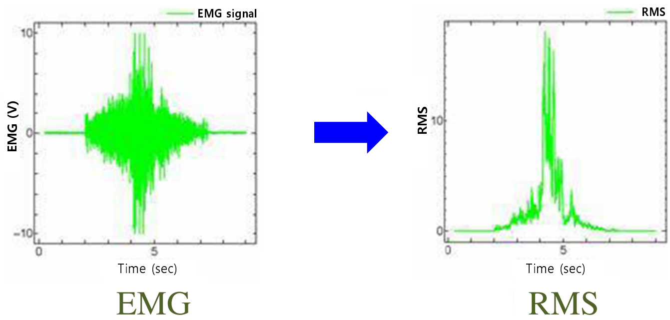

Fig. 1. RMS of EMG signal

2.2 Motion Estimation

When certain motion is performed, the EMG signals of the related multiple muscles show the unique patterns. Therefore, human lower-limb motion can be estimated by monitoring EMG signals of certain muscles of the user since the magnitude of the RMS of the EMG signal indicates the activity level of the muscles. The lower-limb motion (i.e., hip joint flexion/extension motion, knee joint flexion/extension motion, and ankle joint flexion/extension motion) can be estimated by monitoring eight channels of EMG signals per leg. The locations of EMG electrodes are shown in Fig. 2. The relationship between the eight RMS values and the generated vector of joint torques is implemented as shown in eq. (2).

$$\begin{bmatrix} \tau_h \\ \tau_k \\ \tau_a \end{bmatrix} = \begin{bmatrix} w_{h1} & w_{h2} & \cdots & w_{h7} & w_{h8} \\ w_{k1} & w_{k2} & \cdots & w_{k7} & w_{k8} \\ w_{a1} & w_{a2} & \cdots & w_{a7} & w_{a8} \end{bmatrix} \begin{bmatrix} ch_1 \\ ch_2 \\ \vdots \\ ch_7 \\ ch_8 \end{bmatrix} \tag{2}$$

where τ_h, τ_k, and τ_a are torques for hip flexion/extension, knee flexion/extension, and ankle flexion/extension, respectively. ch_j represents the RMS value of the

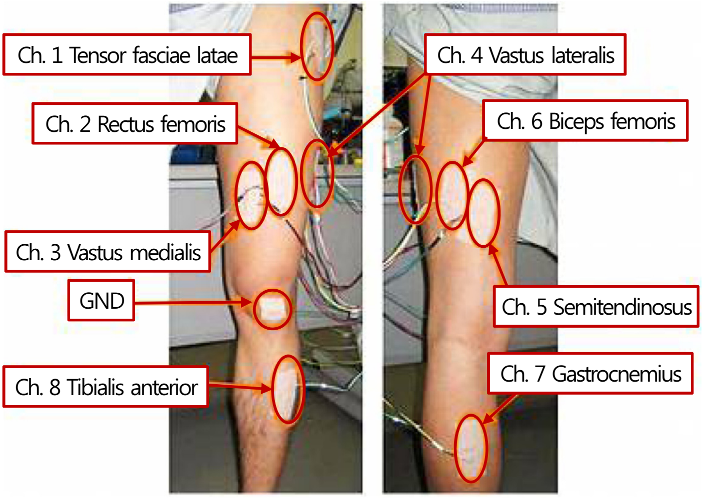

Fig. 2. Location of each electrode

EMG signal measured in channel i. w_{hj} is the weight value for j^{th} EMG signal to estimate the hip flexion/extension motion, w_{kj} is the weight value for j^{th} EMG signal to estimate the knee flexion/extension motion, and w_{aj} is the weight value for j^{th} EMG signal to estimate the ankle flexion/extension motion.

Practically, it is not easy to define the proper weight matrix from the beginning even with the knowledge of human anatomy or the experimental results. Furthermore, the posture of the lower-limb affects the relationship between the EMG signals and the generated joint torques because of anatomical reasons such as change of the moment arm. In other words, the role of each muscle for a certain motion varies in accordance with joint angles. Consequently, the effect of the posture difference of the lower-limb must be taken into account to estimate the correct lower-limb motion for the power assist.

2.3 Neuro-fuzzy Modifier

The neuro-fuzzy modifier has been proposed in order to take into account the effect of the posture difference for the motion estimation [12]. The neuro-fuzzy modifier is used to adjust the weight matrix in eq. (2) by multiplying the coefficients in accordance with the lower-limb posture of the user, so that the effect of lower-limb posture difference can be compensated effectively. It also makes the same effect of adjusting the weight matrix (i.e., the muscle-model matrix) itself to be suitable for each person. The structure of the neuro-fuzzy modifier is the same as a neural network and the process of the signal flow in the neuro-fuzzy modifier is the same as that in fuzzy reasoning.

The architecture of the neuro-fuzzy muscle-model modifier is depicted in Fig. 3. Here, Σ means the summation of the inputs and Π means the multiplica-

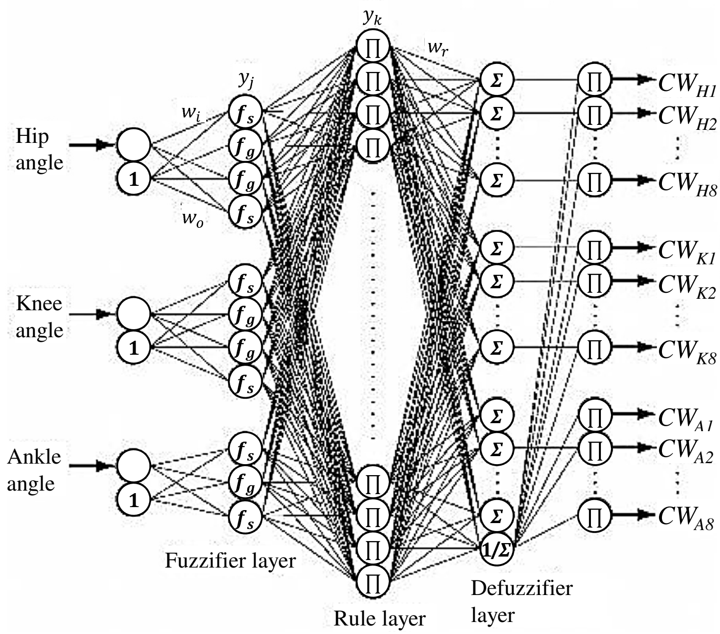

Fig. 3. Nuero-fuzzy modifier

tion of the inputs. The neuro-fuzzy modifier consists of five layers (input layer, fuzzifier layer, rule layer, defuzzifier layer, and output layer). Two joint angles (hip flexion/extension angle and knee flexion/extension angle) are used as inputs for the neuro-fuzzy modifier. Each joint angle is divided into three regions (i.e., FL: flexed region, IM: intermediate region, and EX: extended region for hip flexion/extension angle and knee flexion/extension angle. The output form the neuro-fuzzy modifier is used as a coefficient for each weight of the muscle-model matrix in eq. (2). Consequently, the weight matrix in eq. (2) is adjusted by the neuro-fuzzy method according to the user's lower-limb posture at every sampling time during the control.

In the fuzzifier layer, the degree of fitness of each joint angle is sent to the rule layer. Two kinds of nonlinear functions (f_G: Gaussian function and f_S: Sigmoidal function) are used as the membership functions of the neuro-fuzzy modifier.

$$f_s(u_s) = \frac{1}{1+e^{-u_s}} \tag{3}$$

$$u_s(x) = w_0 + w_i x \tag{4}$$

$$f_G(u_G) = e^{-u_G^2} \tag{5}$$

$$u_G(x) = \frac{w_0 + x}{w_i} \tag{6}$$

where x is the input signal, w_0 is a threshold value and w_i is a weight. The rules for every combination of the joint angle are prepared in the rule layer. The output of the neuro-fuzzy modifier is calculated by dividing the weighted summation of the degree of fitness of each rule (i.e., the output from the rule layer) by the summation of the degree of fitness of each rule. The initial weight for each rule is set to be 1.0, so that the coefficient for every weight in eq. (2) is 1.0 at first.

The neuro-fuzzy muscle-model modifier adapts itself to each user by performing the training with the information of the force sensors. If there is motion difference between the user's lower-limb motion and the exoskeleton's motion, the force sensors detect the force between them. If the weight matrix is perfectly defined, there is no motion difference between the user's lower-limb motion and the exoskeletons motion. Therefore, if the neuro-fuzzy muscle-model modifier is trained to eliminate the force between the users lower-limb and the exoskeleton, the weight matrix becomes perfect even though the initially defined values are not correct. The error back-propagation learning algorithm is applied to minimize the force between the users lower-limb and the exoskeleton. The equation of the evaluation function is written as:

$$E = \frac{1}{2} f_{err}^2 \tag{7}$$

where E is the error function to be minimized and f_{err} is the measured force/ torque between the user and the exoskeleton robot. The lower-limb motion of the user would be properly estimated in real-time after the training the neuro-fuzzy muscle-model modifier.

By the effect of the neuro-fuzzy modifier, w_{hj}, w_k, and w_{aj} are adjusted as follows:

$$\begin{aligned} w_{hj} &= w_{h0,j} \times CW_{Hj} \\ w_{kj} &= w_{k0,j} \times CW_{Kj} \\ w_{aj} &= w_{a0,j} \times CW_{Aj} \end{aligned} \tag{8}$$

where w_{h0j}, w_{k0j}, and w_{a0j} are the initial weight values. CW_{Hj}, CW_{Kj}, and CW_{Aj} are the output from the neuro-fuzzy modifier which are changed according to the users lower-limb posture in order to adjust the relationship between each joint torque and the RMS values in real-time.

3 Power-Assist

3.1 Lower-Limb Power-Assist Robot

The architecture of the exoskeleton robot is shown in Fig. 4. The exoskeleton robot consists of a waist holder, a thigh holder, a lower leg holder, two DC motors

(Maxon DC Motors), two links, a footrest and two force sensors (hip and knee force sensors) in one leg. Locations of force sensors are shown in Fig. 1. There is a passive DOF (dorsiflexion/ planterflexion) in ankle joint and two active DOF (flexion/extension) in hip and knee joints. Each DC motor generates the assist torque at each joint.

Usually, the limitation of human knee movable range is 150 degrees in flexion and 0 degree in extension, and the limitation of human hip movable range is 110 degrees in flexion and 30 degrees in extension. Considering the practical application to everyday life and safety of users, the hip motion limitation of the exoskeleton robot is 110 degrees in flexion and 20 degrees in extension, and the knee motion limitation of the proposed robot is 150 degrees in flexion and 0 degree in extension. The stopper is attached for each joint motion to prevent the exceeding of the movable range. Furthermore, maximum torque of the robot is limited by the hardware and the software for safety.

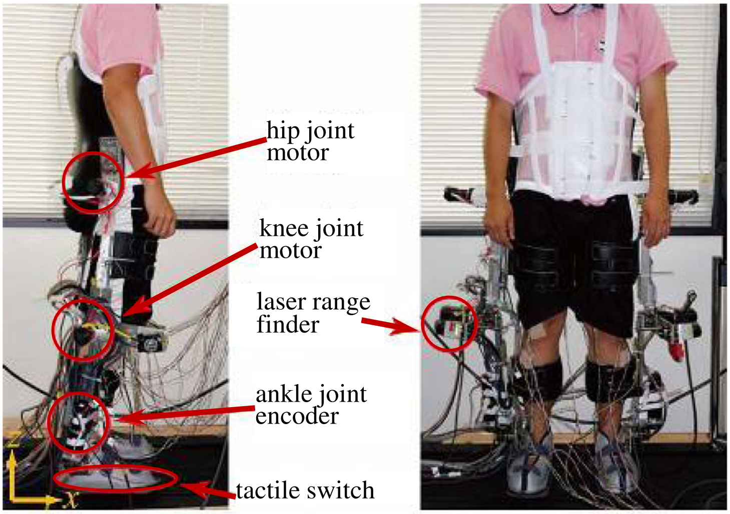

Fig. 4. Lower-limb power-assist exoskeleton robot

3.2 Robot Controller

The controller of the robot consists of the power-assist part and the perception-assist part. Each motor torque is the sum of calculated torques in each part. In this section, the power-assist part is explained in detail. In the power-assist part, the robot is controlled based on two kinds of controller: the force sensor based controller and the EMB-based controller. When the amount of the EMG signals of the user is small, the force sensor based controller is applied to avoid disturbing the users motion. When the amount of the EMG signals of the user is increased, the controller is gradually switched from the force sensor based controller to the EMG-based controller. An example of the membership functions used to switch the signals is shown in Fig. 5. Thus, as a monitored muscle increases the activity level, the controllers are gradually switched from the force sensor based controller to the EMG-based controller.

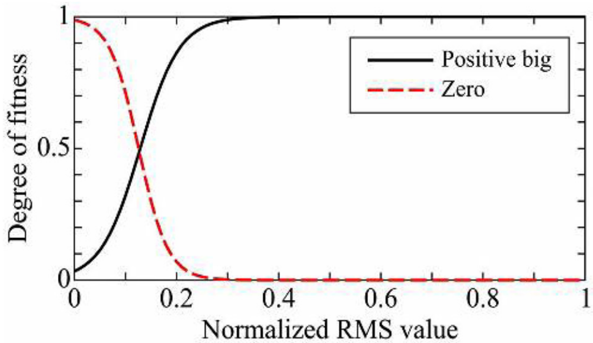

Fig. 5. Membership functions

3.3 EMG-Based Control

The lower-limb motion of the user can be estimated based on the EMG signals as explained in Section 2. Then proper power-assist level can be set by the user to realize the power-assist in accordance with the motion intention of the user.

In order to estimate users motion intention, hand force vector calculation is carried out based on the estimated joint torque vector. The estimated joint torque vector is transferred to the hand force vector of the user using the Jacobian matrix as follows:

$$F_{foot} = J^{-T}\tau_{est} \tag{9}$$

$$F_{f,avg} = \frac{1}{N}\Sigma_{k=1}^{N_f} F_{foot}(k) \tag{10}$$

where F_{foot} is the foot force vector of the user, J is the Jacobian matrix, $F_{f,avg}$ is the average of F_{foot} in N_f number of samples. Then, the desired foot acceleration vector is calculated from eq. (10).

$$\ddot{X}_d = M^{-1}F_{f,avg} \tag{11}$$

where $\ddot{X}_d$ is the desired foot acceleration vector, M is the mass matrix of the robot and the user's lower-limb. To realize the user's intended motion, the following impedance control is applied to obtain the resultant foot force vector F.

$$F = M\ddot{X}_d + B(\dot{X}_d - \dot{X}) + K(X_d - X) \tag{12}$$

where $\dot{X}_d$ d and X_d are the desired hand velocity vector and position vector which are calculated from eq. (11), respectively. B and K are the viscous coefficient matrix and the spring coefficient matrix, respectively. Since impedance parameters of human lower-limb are changed based on the lower-limb posture and the relationship between agonist and antagonist muscles, impedance parameters of the exoskeleton robot are also changed based on them in order to realize natural and comfortable power-assist. Therefore, the impedance parameter matrix B and K in eq. (12) are changed depend on the lower-limb posture and activity levels of activated lower-limb antagonist muscles in real time. Consequently, the joint torque command vector for the joint DC motors is calculated as follows:

$$\tau_{motor} = \kappa J^T F \tag{13}$$

where τ_{motor} is the joint torque command vector, and κ is the power-assist rate.

4 Perception-Assist

4.1 Motion Modification

In the perception-assist part, the robot monitors whether the users motion is in danger or not. Basically the users motion is assisted according to his/her motion intention by the robot. However, if the user has problems in the sensory ability to perceive the environment, the user might not be able to recognize the danger in his/her motion properly. Therefore, if the robot judges the users motion is dangerous, the robot needs to modify the users motion automatically in real-time. This kind of automatic motion modification based on the interaction between the user and the environment is defined as perception-assist. In order to realize the perception-assist, the robot is required to understand not only the motion of the user but also the interaction between the user and the surrounding environment. In the case of lower-limb power-assist robots, the robot needs to prevent the user from falling down during walking, ascending, descending, or standing up. Especially, a user who has problems in the sensory ability to perceive the environment might overlook or misunderstand a bump and stumble on it. Although a healthy person might be able to recover the balance, a user who needs the help of the power-assist robot might fall down. For this reason, the perception-assist is important for the user to avoid an unexpected accident as well as the power-assist.

When the robot detects a the users foot is going to collide with a bump in front of the user, the robot automatically tries to prevent the user from stumbling and falling down by the perception-assist during walking. In this case, the robot tries to modify the users motion by adding the additional modification force in addition to the ordinary power-assist force. Thus, the motion modification is automatically generated regardless of the users intention. For this reason, the user might lose his/her balance and fall down by the effect of the additional modification force of the perception-assist. To prevent such case, the robot takes into account ZMP in the perception-assist algorithm.

A stereo camera or a laser range finder can be used to recognize the environment in front of the user. After a bump in front of the user is detected by the laser range finder or the stereo camera, the distances from the users toe to the bump and the height of the bump are calculated. The robot judge the users toe is going to collide with the bump or not. If the robot find out the users toe is going to collide with the bump, the foot trajectory of the user is automatically modified by adding the additional modification force as well as the power-assist force.

4.2 Examples

In order to prevent the user from losing his/her balance and falling down by the effect of the additional modification force by the perception-assist which is given regardless of the users intention, the robot takes into account ZMP. The supporting leg is found by tactile switches located on both soles. Then, the posture of the users body region is calculated using the joint angles of the supporting leg. ZMP is calculated is written as:

$$x_{zmp} = \frac{\Sigma_{i=1}^{n} m_i(\ddot{z}_i + g)x_i - \Sigma_{i=1}^{n} m_i \ddot{x}_i z_i}{\Sigma_{i=1}^{n} m_i(\ddot{z}_i + g)} \tag{14}$$

where x_{zmp} is the coordinate of ZMP in the horizontal direction, m_i is the mass of each human body part and robot, x_i is the position of the center of gravity (COG) of each part in the horizontal direction, z_i is the position of the COG of each part in the vertical direction and g is the acceleration of gravity. The perception-assist is performed considering x_{zmp}.

When the robot finds out there is no bump in front of the user, the robot evaluates the user's walking based on x_{zmp} in order to prevent the user from falling down in addition to the ordinary power-assist. If x_{zmp} is located in an undesired area, the robot automatically generates additional hip joint torque to change the location of x_{zmp} to the safe area by activating the upper body of the user. The area in which the robot generates the additional hip joint torque in addition to the power-assist is defined as shown in Fig. 6, so that x_{zmp} is moved to center of the support polygon. Since the mass of the upper body region is larger than other regions and the movement of the mass of the upper body is the most effective way to change ZMP, the additional hip joint torque (i.e., the perception-assist torque) is generated at the hip joint motor of the support leg. In Fig. 6, τ_{max} is the amount of the maximum hip joint torque which is defined to protect the user from the injury. The perception-assist torque for walking-assist is calculated using eq. (16).

$$\tau_{walk} = \begin{cases} 0, \ (|\Delta x_{zmp}| < d_{safe}) \\ \frac{\tau_{max}}{d_{cri}}(\Delta x_{zmp} - Sgn \times d_{safe}), \ (d_{safe} \le |\Delta x_{zmp}| \le d_{safe} + d_{cri}) \\ Sgn \times \tau_{max}, \ (|\Delta x_{zmp}| \le d_{safe} + d_{cri}) \end{cases} \tag{15}$$

$$\Delta x_{zmp} = x_{c,ZMP} - x_{zmp}$$

$$Sgn = \mathrm{sgn}(\Delta x_{zmp})$$

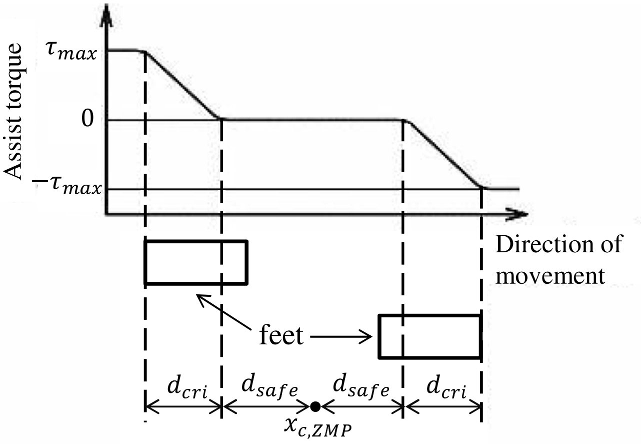

Fig. 6. Perception-assist area based on ZMP

where $x_{c,ZMP}$ is the center position of the support polygon, d_{safe} and d_{cri} are shown in Fig. 6. τ_{max}, d_{safe} and d_{cri} are experimentally defined. When the robot detects a bump, a virtual wall area is generated in front of the bump as shown in Fig. 7(a). The virtual wall area is shown by the trapezoid represented by Δl, Δh and H in Fig. 7 (a). Here, H means the height of the bump, Δl, Δh are the experimentally defined parameters as shown in Fig. 7(a). In the experiments to define Δl and Δh, some subjects walk at normal speed (about $1.4m/sec$). Then, Δl and Δh are defined as minimum values in which the subjects could overcome the bump. If the user's foot enters the virtual wall area as shown in Fig. 7(a), the robot judges the user might stumble on the bump and generates the additional motion modification force along the virtual wall in addition to the power-assist force to avoid the collision with the bump. The additional motor torque generated by the effect of the additional motion modification force (i.e., the perception-assist torque) is written as:

$$\begin{bmatrix} \tau_{h,add} \\ \tau_{k,add} \end{bmatrix} = J^{-T} f_{add} \tag{16}$$

where $\tau_{h,add}$ is the additional hip joint torque, $\tau_{k,add}$ is the additional knee joint torque, J is Jacobian matrix and f_{add} is the additional motion modification force generated along the virtual wall. The amount of f_{add} is related with Δl and Δh. The amount of f_{add} was defined based on the change of ZMP and the feeling of the subjects at the same time when Δl and Δh are defined. Note that $\tau_{h,add}$ and $\tau_{k,add}$ are generated without the user's intention. Therefore, another hip joint torque must be added to compensate for the undesired effect of the ZMP change caused by the additional force, so that the user does not lose his/her balance and fall down by the effect of the perception-assist. Based on eq. (14), τ_{cancel} which is the torque to cancel the effect of the change of ZMP is calculated as:

$$f_{body} = \frac{\tau_{h,add}}{d_{t,cog}} \cdot \frac{x_{t,cog} \sin\theta_{sw,H} - z_{t,cog}\cos\theta_{sw,H}}{x_{b,cog}\sin\theta_b - z_{b,cog}\cos\theta_b} - \frac{\tau_{k,add}}{d_{l,cog}} \cdot \frac{x_{l,cog}\sin(\theta_{sw,k} - \theta_{sw,H}) - z_{l,cog}\cos(\theta_{sw,k} - \theta_{sw,H})}{x_{b,cog}\sin\theta_b - z_{b,cog}\cos\theta_b} \tag{17}$$

$$\tau_{cancel} = f_{body} \times d_{b,cog} \tag{18}$$

where $d_{t,cog}$ is the length between hip joint and the COG of the femoral part, $d_{l,cog}$ is the length between knee joint and the COG of the leg part, and $d_{b,cog}$ are the length between hip joint and the COG of the body part. $x_{p,cog}$ and $z_{p,cog}$ are the position of each body part ($p = t$: femoral part, l: leg part, b: body part).

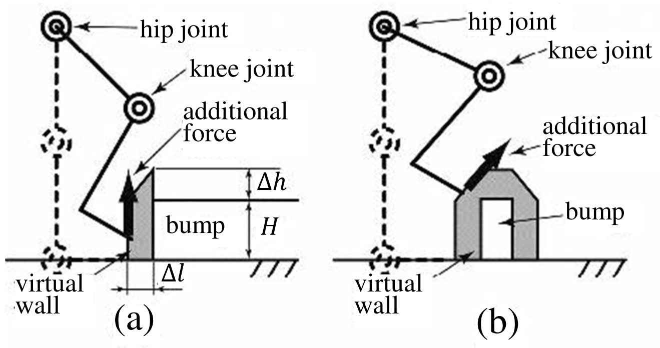

Fig. 7. Virtual wall

After the users foot is lifted over the bump, the robot estimates whether the top surface of the bump has enough space to place the users foot on it. If the top surface of the bump has enough space to place the users foot on it, the robot takes the additional perception-assist torque away from the user and performs the ordinal power-assist. In the case that the top surface of the bump is too narrow, if the robot recognizes the ground ahead on the bump, the robot performs another perception-assist to overcome the bump as shown in Fig. 7(b). Meanwhile, if the robot cannot recognize the ground ahead on the bump, the robot tries to turn back the users foot because the robot cannot find the space to place the users foot on it safety. Thus, the robot takes into account ZMP to prevent the user from losing his/her balance and falling down in any case.

5 Conclusions

In this chapter, an EMG-based control method and a perception-assist method are introduced for the lower-limb power-assist exoskeleton robot. The EMG-based control method is effective method to activate the power-assist robot based on the users motion intention. The perception-assist is also important especially for the user whose sensory ability is deteriorated to secure the safety.

References

1. Guizzo, E., Goldstein, H.: The Rise of the Body Bots. IEEE Spectrum 42(10), 42–48 (2005)
2. Dollar, A.M., Herr, H.: Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art. IEEE Trans. on Robotics 24(1), 144–158 (2008)
3. Yang, C.J., Zhang, J.F., Chen, Y., Dong, Y.M., Zhang, Y.: A Review of Exoskeleton-type Systems and Their Key Technologies. Proc. of IMechE, Journal of Mechanical Engineering Science, Part C 222, 1599–1612 (2008)
4. Rosen, J., Brand, M., Fuchs, M., Arcan, M.: A Myosignal-Based Powered Exoskeleton System. IEEE Trans. on System Man and Cybernetics, Part A 31(3), 210–222 (2001)
5. Kiguchi, K., Kariya, S., Watanabe, K., Izumi, K., Fukuda, T.: An Exoskeletal Robot for Human Elbow Motion Support. Sensor Fusion, Adaptation, and Control. IEEE Trans. on Systems, Man, and Cybernetics, Part B 31(3), 353–361 (2001)
6. Kiguchi, K., Tanaka, T., Fukuda, T.: Neuro-Fuzzy Control of a Robotic Exoskeleton with EMG Signals. IEEE Trans. on Fuzzy Systems 12(4), 481–490 (2004)
7. Kiguchi, K., Esaki, R., Fukuda, T.: Development of a Wearable Exoskeleton for Daily Forearm Motion Assist. Advanced Robotics 19(7), 751–771 (2005)
8. Cavallaro, E.E., Rosen, J., Perry, J.C., Burns, S.: Gravity-balancing leg orthosis and its performance evaluation. IEEE Trans. Robotics 22(6), 1228–1239 (2006)
9. Riener, R., Lünenburger, L., Jezernik, S., Anderschitz, M., Colombo, G., Dietz, V.: Patient-cooperative strategies for robot-aided treadmill training: first experimental results. IEEE Trans. Neural Syst. Rehabil. Eng. 13(3), 380–394 (2005)
10. Hogan, N.: Impedance control: an approach to manipulation, parts I, II, III. J. Dyn. Syst., Meas. Control 107, 1–23 (1985)
11. Blaya, J., Herr, H.: Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait. IEEE Trans. Rehabil. Eng. 12(1), 24–31 (2004)
12. Bar-Cohen, Y.: Electroactive Polymer (EAP) Actuators as Artificial Muscles - Reality, Potential and Challenges. SPIE Press (2004)
13. Noritsugu, T., Tanaka, T.: Application of rubber artificial muscle manipulator as a rehabilitation robot. IEEE/ASME Trans. Mechatronics 2(4), 259–267 (1997)
14. Buerger, S.P., Hogan, N.: Complementary stability and loop shaping for improved human-robot interaction. IEEE Trans. Robotics 23(2), 232–244 (2007)
15. Paluska, D., Herr, H.: Series elasticity and actuator power output. In: Proc. IEEE Int. Conf. Robotics Autom.: ICRA 2006, pp. 1830–1833 (2006)
16. Kong, K., Tomizuka, M.: Flexible joint actuator for patient's rehabilitation device. In: Proc. IEEE Int. Symp. Robot Human Interactive Commun.: ROMAN 2007, pp. 1179–1184 (2007)
17. Pratt, J., Krupp, B., Morse, C.: Series elastic actuators for high fidelity force control. Int. J. Ind. Robot 29(3), 234–241 (2002)
18. Low, K.H.: Initial experiments of a leg mechanism with a flexible geared joint and footpad. Adv. Robotics 19(4), 373–399 (2005)
19. Pratt, G.A., Williamson, M.W.: Series elastic actuators. In: Proc. IEEE/RSJ Int. Conf. Intell. Robotics Syst.: IROS 1995, Pittsburgh, PA, pp. 399–406 (1995)