Making Virtual Walking Real: Perceptual Evaluation of a New Treadmill Control Algorithm

JAN L. SOUMAN
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
PAOLO ROBUFFO GIORDANO
DIS, Università di Roma "La Sapienza", Roma, Italy and
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
ILJA FRISSEN
Max Planck Institute for Biological Cybernetics, Tübingen, Germany and McGill University, Montreal, Canada
ALESSANDRO DE LUCA
DIS, Università di Roma "La Sapienza", Roma, Italy
and
MARC O. ERNST
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

For us humans, walking is our most natural way of moving through the world. One of the major challenges in present research on navigation in virtual reality is to enable users to physically walk through virtual environments. Although treadmills, in principle, allow users to walk for extended periods of time through large virtual environments, existing setups largely fail to produce a truly immersive sense of navigation. Partially, this is because of inadequate control of treadmill speed as a function of walking behavior. Here, we present a new control algorithm that allows users to walk naturally on a treadmill, including starting to walk from standstill, stopping, and varying walking speed. The treadmill speed control consists of a feedback loop based on the measured user position relative to a given reference position, plus a feed-forward term based on online estimation of the user's walking velocity. The purpose of this design is to make the treadmill compensate fully for any persistent walker motion, while keeping the accelerations exerted on the user as low as possible.

We evaluated the performance of the algorithm by conducting a behavioral experiment in which we varied its most important parameters. Participants walked at normal walking speed and then, on an auditory cue, abruptly stopped. After being brought back to the center of the treadmill by the control algorithm, they rated how smoothly the treadmill had changed its velocity in response to the change in walking speed. Ratings, in general, were quite high, indicating good control performance. Moreover, ratings clearly depended on the control algorithm parameters that were varied. Ratings were especially affected by the way the treadmill reversed its direction of motion. In conclusion, controlling treadmill speed in such a way that changes in treadmill speed are unobtrusive and do not disturb VR immersiveness is feasible on a normal treadmill with a straightforward control algorithm.

1. INTRODUCTION

In recent years, virtual reality (VR) has become increasingly realistic and immersive. Both the visual and auditory rendering of virtual environments have been improved significantly, thanks to developments in both hardware and software. In contrast, the possibilities for intuitive navigation through virtual environments (VE) are still relatively rudimentary. Most commonly, users can "move" through high-fidelity virtual environments using a mouse or a joystick. Of course, the most natural way to navigate through VR would be to walk. For small scale virtual environments, one can simply walk within a confined space. The VE can be presented by a cave-like projection system, or by means of a head-mounted display combined with head-tracking. For larger VEs, however, this quickly becomes impractical or even impossible.

In principle, a more general solution is offered by treadmills, as they keep the user in a relatively restricted area while at the same time allowing to walk through arbitrarily large VEs. At present, however, the treadmill solution is still far from satisfactory because of two problems. First, most of the current treadmill set-ups only allow for walking in one direction, severely restricting the possibilities for navigation through VEs. Although several attempts have been made to create an omnidirectional treadmill, allowing users to walk in any direction, most of these solutions at present still are in the prototype state and often are far from perfect [Darken et al. 1997; Iwata 1999; Fernandes et al. 2003; Huang 2003; Nagamori et al. 2005].

The second major problem in using treadmills for VE navigation is how to control the velocity of the treadmill as a function of the walking behavior of the user. Obviously, the user should be kept on the treadmill while either walking or standing still. Most available set-ups, therefore, allow the user to walk at only one fixed speed, given by the treadmill. A better solution would be to have the treadmill respond to changes in walking speed, but this creates its own set of problems. Acceleration of the treadmill cannot be too high, because this would disrupt the immersiveness of the VR or even bring the user out of balance. On the other hand, it cannot be too low either, because then the person would walk off the treadmill when he or she changes walking speed.

This article focuses on the second problem. We developed a control algorithm to control the speed of a treadmill in such a way that VR immersiveness is not disrupted by changes in treadmill velocity [De Luca et al. 2006a; 2006b; 2007]. This algorithm was developed to work with an omnidirectional treadmill, allowing for changes in both walking speed and direction. In this article, we describe a one-dimensional variant of the algorithm and report the results of an experiment that we conducted to evaluate its effectiveness. The algorithm can be used to control a normal, linear treadmill, within the limits of its size and speed. The treadmill will respond smoothly to changes in walking speed, allowing the user to start walking from standstill, to vary walking speed in a natural way, and even to abruptly stop walking without obtrusive changes in treadmill speed.

We had two main objectives in mind when we developed and evaluated the control algorithm. First, we wanted to allow for acceleration and deceleration of the user with respect to the treadmill belt, retaining inertial cues to changes in walking speed. These cues have been shown to be important, both for maintaining postural stability during walking and for the perception of walking speed [Jahn et al. 2000; Mittelstaedt & Mittelstaedt 2001; Cathers et al. 2005; Fitzpatrick et al. 2006]. For this reason, recently described solutions for keeping the user in a more or less constant position with varying walking speeds are unsatisfactory for the purpose of creating a truly immersive virtual-walking environment [Minetti et al. 2003; Lichtenstein et al. 2007]. One approach to solve this problem is to simulate the inertial forces during normal walking while walking in place. This has been done by means of an inertial-force feedback device (a tether), which either pushes or pulls at the back of a person walking on the treadmill [Christensen et al. 2000; Hollerbach et al. 2000]. These studies report that applying the force-feedback improved the degree to which walking on the treadmill was perceived as realistic and natural. The necessary tether forces were mitigated by also allowing for some forward motion of the user on the treadmill. However, one problem with this solution is how to transfer the forces from the force feedback device to the torso of the user. Even with a very stiff harness, force application still differs for accelerations versus decelerations [Checcacci et al. 2003]. In addition, using a mechanic solution for the inertial forces problem makes it more complicated to use for a truly omnidirectional treadmill. Therefore, rather than keeping the user in place and simulating the inertial forces, we decided to allow the user to actually move with respect to the treadmill, keeping the normal inertial cues that occur with changes in walking speed.

Our second objective was to make changes in treadmill speed as smooth as possible and to avoid any jumps in commanded treadmill speed. Fast accelerations of the treadmill would be easily detectable by the user and might lead to instability or loss of balance while walking or standing. Furthermore, if the treadmill is used to navigate through VEs, high accelerations will disrupt the immersiveness of the VR. Our evaluation experiment was specifically targeted at fine-tuning the control algorithm described in the following text in order to minimize the obtrusiveness of changes in treadmill speed. To this end, various parameters of the algorithm were systematically varied.

2. CONTROL ALGORITHM

We decided to allow for normal acceleration and deceleration through space rather than keeping the user in place on the treadmill. Our algorithm allows the user to start walking on a stationary treadmill and the treadmill will only gradually respond to the movement of the user. Similarly, when the user stops walking, the treadmill only gradually comes to a halt and brings the user back to the center of the treadmill. This strategy works best with a relatively large treadmill, but the algorithm can be implemented with treadmills of any size.

The purpose of the control algorithm is to keep the walking person close to a predefined reference position (the center of the treadmill), while keeping changes in treadmill speed as unobtrusive as possible. Formally, this can be described as an output regulation of the position of the user x along the main axis of the treadmill to a reference position x_{ref} in the presence of unknown disturbances (the intended voluntary user walking speed V_w), with v_C as the available treadmill speed command. In developing our control algorithm, we started with a simple first-order control law, that is, by assuming treadmill velocity as the available input. However, in pilot studies that evaluated this control algorithm, we realized that we had to extend this design to a second-order level (with treadmill acceleration as new input) to take into account suitable acceleration bounds in the treadmill speed command. In addition, we decided to use an adaptive reference position, dependent on the walking speed of the user. These three levels of control are described in the following text.

2.1 First-Order Control Law

A first-order kinematic model of the treadmill/user system can be expressed as

$$\begin{cases} \dot{x} = -v_C + V_w \\ \quad y = x \end{cases} , \tag{1}$$

where y stands for the system measurable output, that is, the absolute user position. The sign of v_C in Equation (1) is due to our definition of the direction of walking and treadmill motion: Forward walking (positive change in user position x) corresponds to backwards motion of the treadmill belt (v_C). When the user stands still ($V_w = 0$), exponential stabilization of Equation (1) at a given x_{ref} can be achieved by means of the simple proportional feedback

$$v_C = k_{pos}(x - x_{ref}), \tag{2}$$

where gain $k_{pos} > 0$ tunes the convergence rate. In the presence of persistent walker motion, however, Equation (2) is not able to fully recover the reference position x_{ref}. In particular, if the user walks indefinitely with constant velocity $\bar{V}_w$, at steady state, a position error $\frac{\bar{V}_w}{k_{pos}}$ from x_{ref} will be present. In order to compensate for the walker's locomotion and make the system partially insensitive to changes in walking speed, feedback (Equation (2)) can be complemented with a feed-forward term based on online estimation of walking velocity V_w. Such an estimated value $\tilde{V}_w$ is obtained as output of the scalar dynamic observer system

$$\begin{cases} \dot{\xi} = -v_C + k_{obv}(x - \xi) \\ \tilde{V}_w = k_{obv}(x - \xi) \end{cases} , \quad k_{obv} > 0. \tag{3}$$

Here, variable ξ represents the internal state of the observer that evolves according to the first row in Equation (3). The second row can be seen as the output of the observer, where the estimation of the velocity is computed in terms of the measurement x and the internal state ξ. This observer output is then added to Equation (2) to get the feedback/feed-forward law

$$v_C = k_{pos}(x - x_{ref}) + \tilde{V}_w. \tag{4}$$

Note that, implementation of Equation (3) requires the knowledge of the treadmill velocity v_C besides the user position x. Moreover, from Equations (3) and (1), it follows $\dot{\tilde{V}}_w = k_{obv}(V_w - \tilde{V}_w)$, which can be shown to result in $\tilde{V}_w$ as a low-pass filtered version of V_w with the cut-off frequency tuned by k_{obv}. By analyzing the overall closed-loop system in the Laplace domain, it is possible to prove that (at steady-state) feedback (Equation 4) fully recovers the reference position x_{ref} in presence of any constant disturbance V_w. In practice, the observer acts as an independent dynamical system with its own internal state, designed so that it can asymptotically track the behavior of the unmeasurable intentional velocity $V_w(t)$ of the walker (see Appendix A for further details).

2.2 Second-Order Control Law

Feedback/feed-forward system (Equation (1)) together with observer (Equation (3)) is able to meet the control goals stated at the beginning of this section: output regulation to x_{ref} despite the presence of a persistent disturbance V_w. However, this first-order design of Equations (1) and (4) does not allow any explicit control over the imposed accelerations during transients in walking speed. As stated earlier in the text, it is important to avoid big acceleration steps in treadmill speed (discontinuities in v_C) and to limit the maximum acceleration imposed on the walker in order to allow for truly immersive locomotion through VEs. A convenient way to explicitly address these issues is to perform a dynamic extension of Equation (1) by considering v_C as a state of the system, and its derivative $\dot{v}_C = a_C$ as the

new (acceleration) input signal. Thus, we obtain the extended second-order walker/treadmill system

$$\begin{cases} \dot{x} = -v_C + V_w \\ \dot{v}_C = a_C \\ y = x \end{cases}.$$ (5)

Now, a_C is the new control input and continuity of state v_C is automatically guaranteed, that is, no velocity steps can be commanded. Moreover, bounds on the imposed acceleration can be directly implemented. A number of methodologies for dynamic extension of lower-order control laws have been proposed in the control literature, such as backstepping [Krstić et al. 1995] and the theory of cascaded systems [Panteley and Lorýa 1998]. In the following, we will adopt the latter for our second-order control law design.

Let $v_C^d(t)$ represent the nominal first-order control law (Equation (4)) developed for the original system (Equation (1)). Intuitively, the new command input a_C should impose a perfect tracking of $v_C^d(t)$ on the new state v_C in order to reproduce the same behavior of the first-order scheme. This can be achieved by designing a_C as

$$a_C = \frac{dv_C^d(t)}{dt} + k_a(v_C^d(t) - v_C) = k_{pos}(-v_C + V_w) + \dot{V}_w + k_a(k_{pos}(x - x_{ref}) + \tilde{V}_w - v_C),$$ (6)

where $k_a > 0$ is a suitable gain. According to the theory of cascaded systems, feedback (Equation (6)) is able to stabilize the system (Equation (5)) by achieving an asymptotic tracking of $v_C^d(t)$, tuned by the value of k_a. Note, however, that implementation of Equation (6), requires some quantities that cannot be directly measured, that is, walker velocity V_w, and $\dot{V}_w = k_{obv}(V_w - \tilde{V}_w)$ which again depends on V_w. For this reason, we decided to implement an approximated version of Equation (6), where V_w is replaced by its estimate $\tilde{V}_w$, and $\dot{V}_w$ is neglected. Therefore, the implemented acceleration-level control law is

$$a_C = k_{pos}(-v_C + \tilde{V}_w) + k_a(k_{pos}(x - x_{ref}) + \tilde{V}_w - v_C).$$ (7)

By studying the characteristics of this control law in the Laplace domain, one can verify that closed-loop stability is preserved despite the approximations done in Equation (7) with respect to the nominal control law (Equation 6) (see Appendix A). Note that our setup does not allow a direct acceleration command but only a velocity command. The corresponding velocity command v_C sent to the treadmill can be recovered from Equation (7) by means of a software integration $v_C = \int a_c dt$.

2.3 Tuning of the Reference Position

The largest changes in commanded treadmill speed will occur when the user starts walking from standing still or abruptly stops from walking at normal speed. At steady state, the treadmill velocity v_C exactly matches (with opposite sign) the user velocity V_w. Hence, when the user starts walking, the control must accelerate from $v_c = 0$ to $-V_w$. Vice versa, when the user stops, the control must decelerate from $v_C = -V_w$ to 0. Because the user must be kept within the treadmill boundaries, this poses a lower bound on the necessary acceleration/deceleration. At the same time, the acceleration must be kept low enough not to interfere with the immersiveness of the VR. These two criteria may be incompatible with each other. A possible way to relax the lower acceleration bound is to "virtually" increase the size of the treadmill by changing the reference position x_{ref} according to the user velocity V_w: The faster the user moves, the more x_{ref} is shifted toward the treadmill border in the walking direction of the user. Such behavior can be implemented by defining

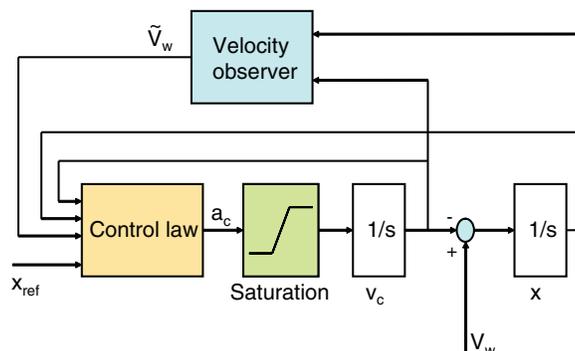$$x_{ref} = k_{ref}\tilde{V}_w + x_{ref,0},$$ (8)

Fig. 1.   Schematic representation of the control algorithm. The Velocity observer estimates the walking velocity V_w of the user based on measured treadmill speed v_C and user position x. The estimated walking speed $\tilde{V}_w$ serves as input to the Control law, together with the measured user position x, the current treadmill speed v_C, and the reference position x_{ref}. The output acceleration a_C is bounded and then integrated to produce the commanded treadmill speed v_C. Note that in this diagram, the commanded treadmill speed v_C equals the measured treadmill speed v_C. In practice, we measured the actual treadmill speed, which could be slightly different from the commanded speed due to the dynamics of the treadmill system.

where $x_{ref,0}$ represents the reference position for a standing user, and k_{ref} is chosen such that when the user moves at maximum speed, x_{ref} does not exceed the treadmill borders. The consequence of this change in control law is that constant disturbances V_w are not completely compensated for anymore, but only attenuated by a factor k_{ref}. Control system behavior for a static user remains unchanged relative to the case with a constant reference position.

2.4   Control Law Implementation

A conceptual scheme of control system (Equation (5)) with control law (Equation (7)), reflecting the actual implementation we used, is shown in Figure 1. The control system consists of two main components: the velocity observer and the control law. In addition, the acceleration command is subject to a saturation function. The purpose of this saturation was to implement a hard constraint on the accelerations imposed on the walker. The reference position x_{ref} was made dependent on the walking speed, as described earlier in the text (not shown in the Figure).

The different parameters in Equation (Equation (7)) have different effects on the behavior of the treadmill in response to the walking behavior of the user. In particular, the observer gain k_{obv} tunes the responsiveness of the system to changes in walking speed V_w. The parameters k_{pos} and k_a will both affect how fast the user is pulled back to the reference position x_{ref} (see Appendix A). In the experiment described in the following text, we varied parameters k_{pos} and k_{obv}, and the acceleration bound a_{max}. As k_{pos} and k_a have similar effects, we only varied the former while keeping the latter constant.

3.   EXPERIMENTAL EVALUATION

The experiment served, first of all, to evaluate the usefulness of the algorithm, and secondly, to systematically test the effects of different parameter values on the extent to which users felt they could walk normally on the treadmill.

Obviously, if one wants to allow for the entire spectrum of normal human walking behavior in a VE, a whole range of scenarios is thinkable for evaluating the control algorithm. In pilot studies, we tested the algorithm with: (i) starting to walk on a stationary treadmill; (ii) abruptly stopping to walk on a moving treadmill; (iii) walking at a constant speed, and (iv) walking at greatly varying speeds. From these experiments, we observed that the most demanding situation for treadmill control is the one where the

Fig. 2. Treadmill setup. Participants walked freely on the treadmill, wearing a safety harness that was connected to a rail above the treadmill. The safety harness prevented them from walking off the treadmill or falling and also helped them to walk straight-ahead while blindfolded. Participants wore a helmet with reflective markers, which were used to track the position of the head.

user abruptly stops when walking at normal walking speed. This concurs with other experiments in which we measured the magnitude of changes in treadmill speed that people can detect while walking or standing. These thresholds were found to be much lower during standing still than during walking on a treadmill (unpublished data). Therefore, we chose to use this stopping-from-walking scenario for the evaluation of our treadmill control algorithm.

3.1 Methods

3.1.1 *Participants*. Thirteen people (5 male, 8 female) participated in the experiment. Their ages ranged from 22 to 31 years. The participants were naïve with respect to the purpose of the experiment.

3.1.2 *Apparatus*. The experiment was conducted on a large custom-made treadmill (Bonte Technology, Zwolle, the Netherlands; see Figure 2). It measured 6.0m (l) $\times$ 2.4m (w) and was capable of speeds up to 40km/h, with a maximum acceleration of 3.0m/s^2. Speed commands to the treadmill were given over RS232 serial connection by custom written software (command rate 30Hz, which was adequate for online treadmill control). Actual treadmill speed was read out over the same serial connection from the treadmill controller. A dynamic characterization of the treadmill mechanics is given in Appendix B. For safety purposes, participants wore a safety harness that was attached to a cable running above the treadmill, which prevented them from walking off the treadmill or from falling down. To mask the sound of the treadmill, participants wore headphones, which played white noise during the experiment. Moreover, they were wearing earplugs to attenuate all auditory input, and were blindfolded. Masking

noise and earplugs together ensured that participants did not received auditory cues to changes in treadmill speed.

The position of the participant on the treadmill was measured with a Vicon optical infrared tracking system (Vicon, Oxford, UK). It tracked reflective markers that were mounted on a helmet worn by the participant (sampling rate 120Hz). From the marker positions, the position of the head of the participant was computed. Together with the actual treadmill speed provided by the treadmill, this formed the input into the control algorithm.

3.1.3 *Procedure.* As indicated previously in the text, we varied three parameters of the control algorithm: the gain k_{obv} with which the walking speed was estimated (0.3, 0.6, or 0.9), the position gain k_{pos} (0.5, 1.0, or 1.5), and the maximum acceleration a_{max} commanded to the treadmill (1.0 or 2.0m/s^2). The appropriate ranges were established in pilot studies. All three parameters were combined factorially, resulting in 18 different conditions. All conditions were replicated five times, in random order. The first replication was considered training and these data were not used in the analysis. Before the experiment, the participants were allowed to walk on the treadmill with vision, experiencing the response of the treadmill to their walking behavior. Experimental trials were performed while blindfolded to ensure that participants would not use visual cues to changes in treadmill speed to base their judgments on.

Each trial started with the participant standing still on the reference position, while the treadmill was stationary. After hearing a beep via the headphones, the participant started to walk forward normally at his/her preferred speed. The safety harness provided a reference to walk straight. During this period, the treadmill responded to the walking behavior with a default set of parameter values that was used in all trials (k_{obv} = 3.0; k_{pos} = 0.5; a_{max} = 5.0). After 6s, a second beep indicated the participant to stop walking and stand still on the moving treadmill. From this point on, treadmill behavior was governed by the parameter values of the condition tested. After the control algorithm had brought the participant back to the reference position and the treadmill was stationary again, a third beep prompted the participant to give an oral rating of treadmill behavior. The rating had to reflect the degree to which the change in treadmill speed was noticeable from the point where he or she stopped walking up to the last beep. The participant was instructed to give a rating from 1 to 10, where 1 represented the case where he or she could not maintain balance and fell (which never happened) and 10 indicated that no change in treadmill motion was felt at all. The ratings were entered into the computer by the experimenter.

4. RESULTS

The ratings given by the participants were averaged across replications of the same condition by taking the median value per participant. A univariate repeated measures ANOVA (SPSS 15.0) was performed to test for significance of the effects of the three parameters (k_{obv}, k_{pos}, and a_{max}). The different parameter values systematically caused the participants to judge the treadmill behavior differently. Figure 3 shows the average ratings as a function of the three parameters that were varied. The error bars in this figure represent the 95%-confidence intervals of the mean responses across participants, after correcting for interparticipant differences. Both the gain of the estimated walking velocity k_{obv} and the position gain k_{pos} had significant effects on the ratings that the participants gave (F(1.38, 16.56) = 9.93, p = 0.001, after Greenhouse-Geisser correction for asphericity, and F(2,24) = 18.99, p < 0.001, respectively). In general, the ratings were higher for lower values of these parameters. The acceleration limit a_{max} did not have a significant effect (compare the left-hand graph in Figure 3 with the right-hand graph; F(1,12) = 1.71, p = 0.215). This probably was because accelerations rarely were larger than 2.0m/s^2. None of the interactions between the three parameters were significant.
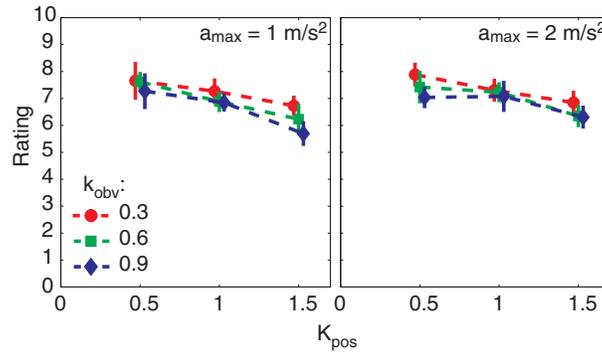
Fig. 3.   Average ratings of treadmill behavior as a function of maximum acceleration (left-hand panel: 1m/s^2; right-hand panel: 2m/s^2), position gain k_{pos} (horizontal axes) and estimated walking velocity gain k_{obv} (color coded lines). For clarity, the different lines have been offset slightly in the horizontal direction. Error bars represent the 95%-confidence intervals of the mean ratings across participants, after correcting for the mean differences between participants. For each participant, the mean rating was subtracted from all the ratings and then the overall mean across all participants was added. This way, the ci's reflect the within-participant variability, rather than between participants.
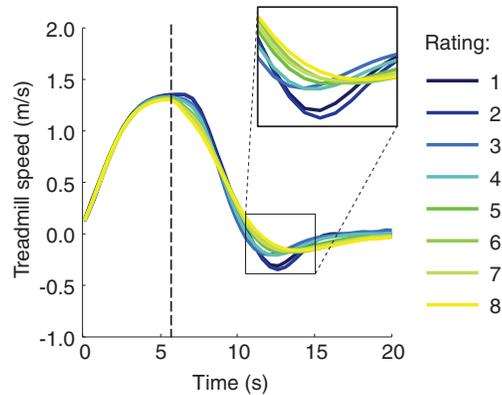

Fig. 4.   Treadmill speed profiles averaged across all trials with the same rating in 0.5s bins, for one exemplary participant. The vertical dashed line indicates the moment at which the participant was instructed to stop walking (after 6s). The inset magnifies the part where the treadmill came to a stop and then reversed its direction.

To analyze in more detail how the different parameter values caused the treadmill to respond differently to the walking behavior of the participants, we computed the average treadmill speed profiles during all trials as a function of the ratings that the participants gave. Trials were first ordered according to the participant's rating. Per rating, we then binned the treadmill speed profiles in 0.5s bins and averaged the treadmill speed for all trials that produced that rating. The average treadmill speed profiles for one exemplary participant are shown in Figure 4. These profiles are similar to those found in most other participants. For each trial, time started at 0s when the participant started to walk. As can be seen from the figure, the treadmill started off at 0m/s, and then increased its speed during the first few seconds of the trial in order to compensate for the walking speed of the participant. As soon as the participant stopped walking, after 6s, the treadmill also decelerated. Normally, the treadmill would overshoot the reference position, come to a stop and then slowly return the person to the reference position by moving forward. The speed profiles show that the biggest difference between trials with

different ratings was in the part where the treadmill came to a stop and then slowly moved forward again. The lowest ratings were given when this overshoot was large; ratings were higher when the overshoot was small and the change in treadmill speed more gradual.

5. DISCUSSION

The main goal in developing the control algorithm was to ensure smooth and gradual changes in treadmill speed in response to changes in walking speed. This is an important prerequisite for immersive navigation by locomotion through VEs. The algorithm developed here is based on both velocity and acceleration control. Effectively, it tries to minimize the distance between the person walking on the treadmill and a given reference position. At the same time, it compensates for voluntary motion of the user by means of an online estimation of the walking speed. These two components resemble algorithms suggested before (e.g., by Moghaddam and Buehler [1993], for the control of a robot locomotion set-up, and by Noma and Miyasato [1998] and Noma et al. [2000]). However, our algorithm differs from these control schemes in several important respects.

First, we propose an acceleration-level design, which automatically avoids jumps in the commanded treadmill velocity. This allows explicit limitation of the imposed accelerations. In addition, the reference position to which the system tries to bring the user back to is dynamically adapted, depending on the walking speed of the user.

A second difference is that our control algorithm relies on a continuous estimation of the walker's position that does not require a model of the walking patterns nor measurements of the movements of the feet or legs. Any position measurement system can be used. In the present study, we have used an infrared optical tracking system to track the position of the head, but we have also successfully implemented the algorithm with a video-based markerless tracking system. When using a treadmill for VR purposes, head tracking will probably already be available for visualization. Our algorithm can then be used to control treadmill speed without any additional requirements.

Third, the estimation of the walker's velocity is based on a dynamic observer, and not on static computations as in the article by Noma and Miyasato [1998]. This is implemented by adding a feed-forward term to the feedback action, making it possible to compensate for any walking behavior. Thanks to this effective feed-forward term, we do not have to resort to an integral action in the feedback part of the controller, thus avoiding potential instability or overshooting problems. In addition, this gives more freedom in tuning the gains of the control law.

In the experiment presented here, we evaluated the behavior of the treadmill for different values of the most critical parameters of the control algorithm. In general, higher values for the two gain parameters k_{obv} and k_{pos} that we tested make the treadmill more responsive to changes in walking speed. The downside of this is that, as our results show, the changes in treadmill speed become much more noticeable to the user. Consequently, one should set these values as low as possible, given the dimensions of the treadmill that is being used. If the values are too low, the treadmill will respond too slowly and the user will not be kept within the bounds of the treadmill. Our large treadmill (6m) allowed us to use low values, making changes in treadmill speed small and smooth enough to be almost unnoticeable by the user. On our large scale treadmill, users can even stop walking, turn around, and walk in the opposite direction, without instability or loss of balance. Importantly, users walked without any support for stability. On smaller treadmills, higher values for the gains have to be used, making changes in treadmill speed more noticeable.

The settings of the parameter values become more critical with higher walking speeds. Some of the participants in our experiment walked fairly slowly (<1m/s) and consequently did not show strong effects of the different parameter values. Partially, their slower walking speed may have been due to apprehension of walking blindfolded. Walking with vision through a VE might make it easier for

them to walk at a normal walking speed. On the other hand, adding a VE might also create a conflict between the perceived visual self-motion through the VE and the felt motion through the real world on the treadmill. For instance, when the user stops walking, the visual environment will be stationary. At the same time, the user is moved through space by the treadmill in order to bring him/her back to the reference position. Whether these conflicts are large enough to be noticeable and what their effects will be has to be further investigated. Previous studies have shown that humans calibrate their perceived walking speed to the visual optic flow experienced during walking. This suggests that, at least with time, these conflicts do not necessarily disrupt VR immersiveness [Durgin et al. 2005; Mohler et al., 2007; Rieser et al., 1995]. Preliminary studies carried out on our treadmill set-up suggest that adding a visual VE indeed does improve stability during walking.

As indicated in the Introduction, controlling treadmill speed as a function of walking behavior is only half the challenge of creating a truly immersive locomotion interface for VR. The other goal is to create an omnidirectional treadmill. We are currently developing such a treadmill. A two-dimensional version of the algorithm described here will be used for the control of this omnidirectional treadmill [De Luca et al. 2006a; 2006b; 2007]. Obviously, the demands on the control algorithm will be higher in the omnidirectional case, as now not only speed but also walking direction can change. An indication that users are more sensitive to changes in treadmill direction than speed can be found in the results described previously in the text. In our experiment, the most important part of the motion profile of the treadmill proved to be where it reversed its direction (Figure 4). More studies are needed to investigate whether this represents the most extreme case or that users are even more sensitive to intermediate direction changes. The results from the present study are encouraging evidence, however, that controlling treadmill velocity without disrupting the immersiveness of VR is indeed a feasible task.

Appendix A: Stability of the Control System

A.1. Stability of First-Order Control System

Closed-loop behavior of system Equation (1) together with feedback Equation (4) can be conveniently studied in the Laplace domain. As a preliminary step, it is useful to explicitly characterize the transfer function linking $V_w(s)$ to its estimate $\tilde{V}_w(s)$. From Equations (1) and (3), it follows that $\dot{\tilde{V}}_w = k_{obv}(V_w - \tilde{V}_w)$ from which we obtain

$$\tilde{V}_w(s) = \frac{k_{obv}}{s + k_{obv}} V_w(s). \tag{A1}$$

Hence, the estimate $\tilde{V}_w$ results in a low-pass filtered version of V_w with cut-off frequency tuned by k_{obv}. By plugging Equation (4) into Equation (1), and using the expression (A1), the closed-loop behavior of the user position x is described by the Laplace relation

$$X(s) = \frac{k_{pos}}{s + k_{pos}} X_{ref}(s) + \frac{s}{s + k_{obv}} V_w(s) = F_1(s) X_{ref}(s) + F_2(s) V_w(s). \tag{A2}$$

Therefore, one can verify that, at steady state, any constant disturbance $V_w(s) = \frac{\bar{V}_w}{s}$ is fully compensated thanks to the zero in $s = 0$ of $F_2(s)$, that is, the term $F_2(s) \cdot \frac{\bar{V}_w}{s} = \frac{\bar{V}_w}{s + k_{obv}}$ exponentially vanishes over time for any $\bar{V}_w$. Moreover, any constant reference position x_{ref} is also correctly reproduced at

steady state, since $F_1(s)\big|_{s=0} = 1$, that is, $F_1(s)$ has an unitary steady-state gain. This, together with the rejection properties of $F_2(s)$, ensures that $\lim_{t\to\infty} x(t) = x_{ref}$ for a person who stands still or walks at a constant speed.

A.2. Stability of Second-Order Control System

Stability of the closed-loop behavior of (Equation (5)) together with feedback (Equation (7)) can be again analyzed in the Laplace domain. By plugging Equation (7) in Equation (5), and using Equation (A1), we get the closed-loop transfer functions

$$
\begin{aligned}
X(s) &= \frac{k_{pos}k_a}{(s+k_{pos})(s+k_a)}X_{ref}(s) + \frac{s(s+k_{obv}+k_{pos}+k_a)}{(s+k_{obv})(s+k_{pos})(s+k_a)}V_w(s) = \\
&= F_3(s)X_{ref}(s) + F_4(s)V_w(s),
\end{aligned}
\tag{A3}
$$

which are stable (poles with negative real part) as long as $k_{pos} > 0$, $k_a > 0$ and $k_{obv} > 0$. Hence, stability is guaranteed by imposing positive gains, in contrast to, for example, Noma and Miyasato [1998], where the control gains are restricted to a specific stability region. Our implementation allows better tuning of the overall control law.

Note that gains k_a and k_{pos} appear symmetrically in Equation (A3), and in particular, they equally affect the transfer function $F_3(s)$. As in the first-order case, the zero in $s = 0$ of $F_4(s)$ provides a complete steady-state rejection of any constant disturbance V_w, and a constant reference position x_{ref} is correctly reached since it is, again, $F_3(s)\big|_{s=0} = 1$. Consequently, the approximations made in controller (Equation (7)) with respect to the ideal control law (Equation (6)) do not affect the closed-loop stability of the system, and feedback law (Equation (7)) fully meets our goals.

APPENDIX B: DYNAMIC CHARACTERIZATION OF THE TREADMILL SYSTEM

The treadmill control architecture consists of two layers: The high-level control law, as described in Equations (5), (7), and (8), and the low-level controller of the treadmill motor. The high-level feedback action is computed on a standard PC and sent as a reference velocity signal to the low-level direct controller, which is responsible for its realization. Due to the platform dynamics and the presence of communication delays, possible mismatches between the commanded $v_C(t)$ and actual $v_{tm}(t)$ treadmill velocity can arise. On the other hand, for a kinematic control design, one assumes that these two quantities match perfectly, that is, that the velocity output of the PC-based controller is instantaneously executed by the treadmill actuation.

In order to validate this simplifying assumption, we identified experimentally the transfer function between the commanded and actual treadmill velocity, quantifying the system bandwidth in a linear domain. For this, we have collected treadmill velocity data during normal treadmill operation (any controller, even untuned, is suitable to this end) and processed them using standard model identification algorithms (the *pem* function of the MATLAB System Identification Toolbox). This produced the following result

$$
\frac{v_{tm}(s)}{v_C(s)} = P(s) = \frac{6.67s + 7.28}{s^2 + 7.526s + 7.28}.
\tag{B1}
$$

The corresponding Bode diagram is shown in Figure (5a).

Several further experimental data sets were used for the validation of this model (the MATLAB compare function computes the model output associated with a given input and compares it with the measured output). Treadmill velocity data were collected for different walking scenarios, such
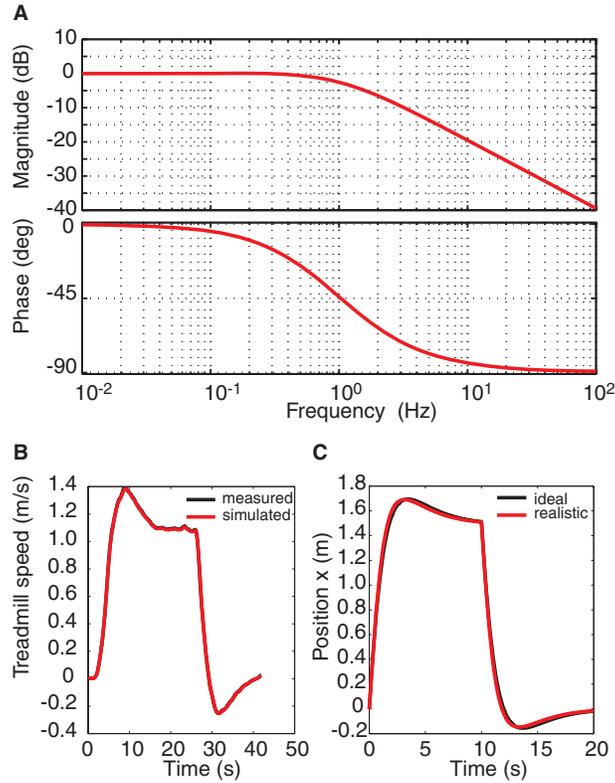
Fig. 5. (A) Bode diagram of the treadmill transfer function (B1), evaluated experimentally from treadmill velocity data collected during treadmill operation. The bandwidth is at 1.1Hz, with a phase lag of 47 deg (equivalent to a time delay of about 120 ms). (B) Validation test of the treadmill transfer function (B1) by comparing actual measured treadmill speed (black) and simulated speed (red). The simulation explained 99.06% of the variance in the actual speed. (C) Simulation of the behavior of user position on the treadmill in a start/walk/stop scenario, using the experimentally derived realistic transfer function (B1, in red) and the ideal transfer function $\frac{v_{tm}(s)}{v_C(s)} = 1$ (black).

as: (i) starting to walk from stand still, (ii) starting to walk followed by walking at an approximately constant speed for an extended period of time (Figure 5b); (iii) changing walking speed, and (iv) abruptly stopping from normal walking speed. In all validation tests, fits higher than 95% (proportion of explained variance) were obtained, confirming the reliability of the identified model (B1). From this model, it follows that the bandwidth of the low-level controller is 1.1Hz, with a corresponding phase lag of $47°$, roughly equivalent to a time delay of 120ms.

We also performed a comparative simulation between the ideal case $\left(\frac{v_{tm}(s)}{v_C(s)} = 1\right)$, where the actual treadmill speed is identical to the commanded speed, and the realistic case that uses the experimentally identified model (B1). The simulation considers a start/walk/stop sequence of the walker, with a constant walking speed of 1.35m/s, held for 10s. The gains of the controller (Equation (7)) were set to $k_{pos} = 0.5$, $k_a = 0.7$, $k_{ref} = 1.6$, and $k_{obv} = 3$. Figures (5c) show the ideal and realistic behaviors of simulated user position on the treadmill. The differences are negligible, allowing us to use the ideal model as an approximation.

REFERENCES

CATHERS, I., DAY, B. L., AND FITZPATRICK, R. C. 2005. Otolith and canal reflexes in human standing. *J. Physiol. 563*, 229–234.

CHECCACCI, D., HOLLERBACH, J. M., HAYWARD, R., AND BERGAMASCO, M. 2003. Design and analysis of a harness for torso force application in locomotion interfaces. In *Proceedings of the EuroHaptics Conference.* IEEE, Los Alamitos, CA, 53–67.

CHRISTENSEN, R. R., HOLLERBACH, J. M., XU, Y., AND MEEK, S. G. 2000. Inertial-force feedback for the tread port locomotion interface. *Presence 9,* 1, 1–14.

DARKEN, R. P., COCKAYNE, W. R., AND CARMEIN, D. 1997. The omni-directional treadmill: A locomotion device for virtual worlds. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology.* ACM, New York, 213–221.

DE LUCA, A., MATTONE, R., AND ROBUFFO GIORDANO, P. 2006a. The motion control problem for the CyberCarpet. In *Proceedings of the IEEE International Conference on Robotics and Automation.* IEEE, Los Alamitos, CA, 3532–3537.

DE LUCA, A., MATTONE, R., AND ROBUFFO GIORDANO, P. 2006b. Feedback/feedforward schemes for motion control of the Cyber-Carpet. In *Proceedings of the IEEE Symposium on Robot Control.* IEEE, Los Alamitos, CA.

DE LUCA, A., MATTONE, R., AND ROBUFFO GIORDANO, P. 2007. Acceleration-level control of the CyberCarpet. In *Proceedings of the IEEE International Conference on Robotics and Automation.* IEEE, Los Alamitos, CA, 2330–2335.

DURGIN, F. H., PELAH, A., FOX, L. F., LEWIS, J., KANE, R., AND WALLEY, K. A. 2005. Self-motion perception during locomotor recalibration: More than meets the eye. *J. Exp. Psych. Human Percept. Perform. 31*, 3, 398–419.

FERNANDES, K. J., RAJA, V., AND EYRE, J. 2003. Cybersphere: The fully immersive spherical projection system. *Comm. ACM 46 ,* 9, 141–146.

FITZPATRICK, R. C., BUTLER, J. E., AND DAY, B. L. 2006. Resolving head rotation for human bipedalism. *Curr. Biol. 16*, 1509–1514.

HOLLERBACH, J. M., XU, Y., CHRISTENSEN, R., AND JACOBSEN, S. C. 2000. Design specifications for the second generation Sarcos Treadport locomotion interface. In *Proceedings of the Haptics Symposium, ASME Dynamic Systems and Control Division.* IEEE, Los Alamitos, CA,69–2, 1293–1298.

HUANG, J.-Y. 2003. An omnidirectional stroll-based virtual reality interface and its application on overhead crane training. *IEEE Trans. Multimedia 5,* 1, 39–51.

IWATA, H. 1999. Walking about virtual environments on an infinite floor. *IEEE Virtual Reality*, 286–293.

JAHN, K., STRUPP, M., SCHNEIDER, E., DIETERICH, M., AND BRANDT, T. 2000. Differential effects of vestibular stimulation on walking and running. *NeuroReport 11,* 8, 1745–1748.

KRSTIĆ, M., KANELLAKOPOULOS, I., AND KOKOTOVIĆ, P. 1995. *Nonlinear and Adaptive Control Design.* John Wiley & Sons, New York.

LICHTENSTEIN, L., BARABAS, J., WOODS, R. L., AND PELI, E. 2007. A feedback-controlled interface for treadmill locomotion in virtual environments. *ACM Trans. Appl. Percept. 4*, 1.

MINETTI, A. E., BOLDRINI, L., BRUSAMOLIN, L., ZAMPARO, P., AND MCKEE, T. 2003. A feedback-controlled treadmill (treadmill-on-demand) and the spontaneous speed of walking and running in humans. *J. Appl. Physiol. 95*, 838–843.

MITTELSTAEDT, M.-L., AND MITTELSTAEDT, H. 2001. Idiothetic navigation in humans: Estimation of path length. *Exp. Brain Res. 139*, 318–322.

MOGHADDAM, M. M., AND BUEHLER, M. 1993. Control of virtual motion systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Los Alamitos, CA, 63–67.

MOHLER, B., THOMPSON, W. B., CREEM-REGEHR, S. H., WILLEMSEN, P., PICK, H. L., AND RIESER, J. J. 2007. Calibration of locomotion due to visual motion in a treadmill-based virtual environment. *ACM Trans. Appl. Percept. 4,* 1.

NAGAMORI, A., WAKABAYASHI, K., AND ITO, M. 2005. The ball array treadmill: A locomotion interface for virtual worlds. *IEEE Virtual Reality* 3–6.

NOMA, H., AND MIYASATO, T. 1998. Design for locomotion interface in a large scale virtual environment ATLAS: ATR locomotion interface for active self motion. In *Proceedings of the 7th Annual Symposium on Haptic Interface for Virtual Environments and Teleoperated Systems.* IEEE, Los Alamitos, CA, 64, 111–118.

NOMA, H., SUGIHARA, T., AND MIYASATO, T. 2000. Development of ground surface simulator for Tel-E-Merge system. In *Proceedings of IEEE Virtual Reality Conference.* IEEE, Los Alamitos, CA, 217–224.

PANTELEY, E., AND LORÝA, A. 1998. On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Syst. Control Lett. 33*, 2, 131–138.

RIESER, J. J., PICK, H. L., ASHMEAD, D. H., AND GARING, A. E. 1995. Calibration of human locomotion and models of perceptual-motor organization. *J. Exp. Psych. Human Percept. Perform. 21*, 3, 480–497.