Acceleration-level control of the CyberCarpet

A. De Luca, R. Mattone, P. Robuffo Giordano

Dipartimento di Informatica e Sistemistica

Università di Roma "La Sapienza"

Via Eudossiana 18, 00184 Roma, Italy

{deluca,mattone,robuffo}@dis.uniroma1.it

Abstract— The *CyberCarpet* is an actuated platform that allows unconstrained locomotion of a walking user for VR exploration. The platform has two actuating devices (linear and angular) and the motion control problem is dual to that of nonholonomic wheeled mobile robots. The main control objective is to keep the walker close to the platform center. We first recall global kinematic control schemes developed at the velocity level, i.e., with the linear and angular velocities of the platform as input commands. Then, we use backstepping techniques and the theory of cascaded systems to move the design to control laws at the acceleration level. Acceleration control is more suitable to take into account the limitations imposed to the platform motion by the actuation system and/or the physiological bounds on the human walker. In particular, the availability of platform accelerations allows the analytical computation of the apparent accelerations felt by the user.

I. INTRODUCTION

The main objective of the European research project *CyberWalk* [1] is to allow omni-directional, unconstrained and natural locomotion to a walker user while exploring virtual worlds.

Different locomotion interfaces already exist that allow walking in virtual environments (see [2] and [3]). However, they usually constrain the feet or the legs of the user. For unconstrained planar walking, the Omnidirectional Treadmill has been proposed in [4] using two perpendicular belts and a large number of rollers, while a torus-shaped belt arrangement is implemented in the Torus Treadmill [5].

A different principle is used in [6], where a conveyor belt and a turntable transmit omni-directional motion to a walker through a ball-array board, realizing thus a 2D treadmill. The motion transmission concept of the *CyberCarpet* platform, under advanced development within the *CyberWalk* project, is similar. Rotating balls are fitted into an array board and are in contact with the belt so that a user on the board moves in the opposite direction of the corresponding point on the belt. The walker is allowed to move (possibly indefinitely) in a natural way and in any direction. The platform controller counteracts her/his motion and pulls the walker toward the center of the *CyberCarpet*. The body pose on the carpet is acquired through a markerless visual tracking system and used to drive two actuation devices that translate and rotate the walker on the platform. The combined walker-platform displacement is needed to update the scene of the virtual environment shown to the user. The overall system architecture is shown in Fig. 1. In our previous work on
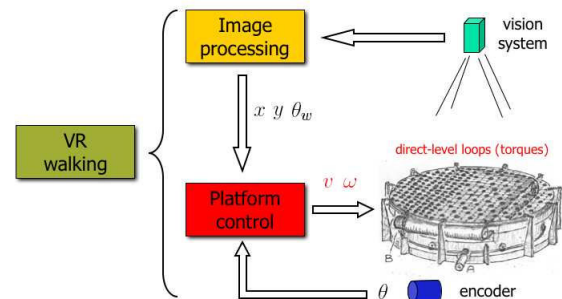

Fig. 1. Control system architecture of the *CyberCarpet*, with a drawing of the preliminary platform design (courtesy of Max Planck Institute for Biological Cybernetics; German Patent filed in 2005)
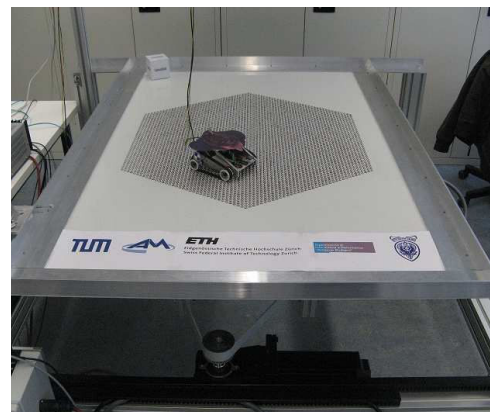

Fig. 2. Picture of the experimental small-scale platform developed at the Institute of Applied Mechanics of the Technical University of Munich

the motion control problem of the *CyberCarpet* [7], we developed a first-order kinematic model of the platform. Recognizing a duality with the control problem for non-holonomic wheeled robots, we proposed a velocity control scheme based on input-output decoupling and linearization. In [8], we proposed then two modified velocity control schemes handling some control singularities affecting the laws in [7], and including also a feedforward term based on an observer of the (unavailable) instantaneous velocity of the

walker. Performance of these control laws was satisfactory also in the preliminary validation tests, see Fig. 2 for a picture of the small-scale prototype of the *CyberCarpet*. However, velocity control design does not allow to take explicitly into account the acceleration bounds imposed by the actuation system and physiologically acceptable for the human walker.

In this paper, we present acceleration-level control schemes for the *CyberCarpet* platform. We consider the extended (second-order) kinematic model of the platform (Sect. II), recall the feedback/feedforward velocity schemes of [7] and [8] (Sect. III), and propose two different techniques (based on *backstepping* and on the theory of *cascaded systems*) for moving the first-order control laws to the acceleration level (Sect. IV). Using the obtained platform acceleration commands, we provide in Sect. V analytical expressions for the inertial, centrifugal and Coriolis components of the apparent acceleration felt by the user. Simulation results are reported in Sect. VI. The evaluation of the overall performance can be further appreciated in the accompanying video clip. Conclusions and future work within the *Cyber-Walk* project are discussed in the final section.

II. KINEMATIC MODELING

A first-order kinematic model of the *CyberCarpet* has been derived in [7]. Given the ball-array surface of the *CyberCarpet*, any actuated motion of the belt will result in a reverse motion imposed to the walker standing on top of the ball array.

With reference to Fig. 3, the walker absolute position and orientation are (x, y) and θ_w, respectively, while θ is the angle of the rotated frame (X_t, Y_t) (with X_t–axis in the direction of the belt along which linear motion is actuated) w.r.t. the absolute frame (X_0, Y_0) (attached to the fixed overlooking camera). When the walker is standing still, we have

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -v\cos\theta + y\omega \\ -v\sin\theta - x\omega \\ \omega \end{bmatrix} = A(x, y, \theta)\begin{bmatrix} v \\ \omega \end{bmatrix}$$
$$\dot{\theta}_w = -\omega,$$
$$(1)$$

being v and ω the linear and angular velocity of the *CyberCarpet*. We refer to [7] for an analysis of the nonholonomic properties of system (1).

For later use, note that the two Cartesian coordinates (x, y) may be replaced by suitable polar coordinates (see also [9]), with $R = \sqrt{x^2 + y^2}$ being the walker's distance from the center and angle $\alpha = \mathrm{atan2}(y, x) - \theta$ locating the position of the walker in the rotated frame (X_t, Y_t). One obtains

$$\begin{aligned} \dot{R} &= -v\cos\alpha \\ \dot{\alpha} &= v\frac{\sin\alpha}{R} - 2\omega. \end{aligned}$$
$$(2)$$
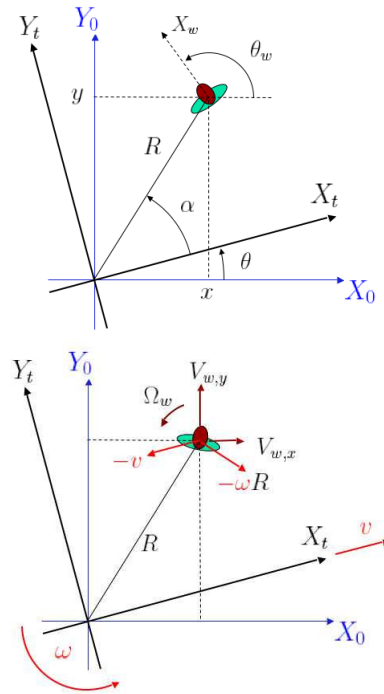
When the walker is in motion, the first-order kinematic

Fig. 3. Frames and variables definition: walker and platform still (top) and in motion (bottom)

model becomes

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = A(x, y, \theta)\begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} V_{w,x} \\ V_{w,y} \\ 0 \end{bmatrix}$$
$$\dot{\theta}_w = -\omega + \Omega_w,$$
$$(3)$$

with absolute linear and angular walker velocities $V_w = (V_{w,x}, V_{w,y})$ and Ω_w, respectively (see bottom of Fig. 3).

When the *CyberCarpet* is commanded by linear and angular acceleration inputs, v and ω become state variables, and the kinematic models (1) and (3) must be extended by the equations

$$\begin{aligned} \dot{v} &= a, \\ \dot{\omega} &= \eta, \end{aligned}$$
$$(4)$$

where a and η will be taken as control inputs.

III. THE CONTROL DESIGN PROBLEM

The control task requires the regulation of the walker position (x, y) to the platform origin in the presence of unknown disturbances (the intended voluntary motion of the walker (V_w, Ω_w)). Actually, ultimate boundedness of the position error to a small target disk around the center could be sufficient, since the practical objective is just to keep the walker safely within the physical boundaries of the platform.

In order to design the linear and angular platform accelerations commands a and η that perform the assigned task, we exploit here the availability of the velocity-level control laws already developed in our previous works [7] and [8], and that have shown a satisfactory performance in the experimental

validation on a prototype of the platform (see `http://www.dis.uniroma1.it/~labrob/research/CW.html` for videos of the experiments). Therefore, we summarize in the following our previous results on first-order kinematic control of the *CyberCarpet*.

A. Velocity-level control

The velocity control design for the platform motion assumes that only the absolute position of the walker is available through a visual tracker, together with the angular orientation of the turntable measured by an encoder.

In [7], we presented a simple control design based on input-output feedback linearization leading to

$$v = \frac{k(x^2 + y^2)}{x\cos\theta + y\sin\theta} = \frac{kR}{\cos\alpha} \qquad (5)$$

and

$$\omega = \frac{k(y\cos\theta - x\sin\theta)}{x\cos\theta + y\sin\theta} = k\tan\alpha, \qquad (6)$$

which are expressed in terms of either the Cartesian errors x and y or the radial and angular errors R and α. When the walker stands still, the control law achieves exponential stabilization of the walker position to the origin, with arbitrary and independent rates for the x and y components. This decoupling control law, however, is affected by singularity when the walker is on the Y_t axis (in particular at the origin).

In order to deal with these control singularities, we proposed in [8] suitable modifications that preserve some convenient characteristics of the decoupling law (5–6). The resulting control law is

$$
\begin{aligned}
v &= kR^2 \operatorname{sgn}(\cos\alpha) \\
&= k(x^2 + y^2)\operatorname{sgn}(x\cos\theta + y\sin\theta),
\end{aligned} \qquad (7)
$$

and

$$
\begin{aligned}
\omega &= kR\sin\alpha\operatorname{sgn}(\cos\alpha) \\
&= k(y\cos\theta - x\sin\theta)\operatorname{sgn}(x\cos\theta + y\sin\theta),
\end{aligned} \qquad (8)
$$

and is now defined at any system configuration. Under the feedback law (7–8) and in the absence of walker locomotion, the closed-loop input-output dynamics becomes

$$\dot{x} = -kR\left|\cos\alpha\right|x, \quad \dot{y} = -kR\left|\cos\alpha\right|y, \qquad (9)$$

which can be proved to be asymptotically stable at the origin by Lyapunov arguments. Although the dynamics (9) is neither linear nor decoupled, it is

$$\frac{y(t)}{x(t)} = \frac{\dot{y}(t)}{\dot{x}(t)} = \frac{y_0}{x_0}, \qquad (10)$$

i.e., the user is pulled toward the origin along the connecting straight line. However, the convergence rate of the x and y variables drops quadratically to zero as the walker approaches the origin.

B. Dealing with walker's velocity

When the walker is in motion, V_w and Ω_w are in general both different from zero and the system first-order kinematics is described by eq. (3). A persistent walker locomotion will in general prevent the convergence of her/his position to the platform center when using the control law (7–8).

In order to deal with walker's locomotion, we have proposed in [8] to use an estimate $\tilde{V}_w$ of the walker linear velocity V_w obtained by a suitable disturbance observer. Based on this estimate, feedforward actions are computed and added to the feedback terms

$$v_c = v + v_f = v + \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}\tilde{V}_w$$

$$\omega_c = \omega + \omega_f = \omega + \operatorname{sat}\left(\frac{1}{R}\begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix}\tilde{V}_w\right). \qquad (11)$$

In (11), $\operatorname{sat}(\cdot)$ is a standard saturation function, with lower/upper saturation limits to be defined according to given constraints on the input velocities. It is readily verified that, for $\tilde{V}_w = V_w$, the feedforward term v_f in (11) compensates for the component of the walker velocity along the direction of the *CyberCarpet* linear motion, while ω_f (in the absence of saturation) cancels the component of V_w in the orthogonal direction[1].

Note that $\tilde{V}_w$ is a low-pass filtered version of the absolute Cartesian velocity V_w, with suitable cut-off frequency k_w to be set in the observer according to the noise level in the measurements of x and y.

IV. ACCELERATION-LEVEL CONTROL

When a control law, with the same regulation objectives stated in Sect. III, has to be devised for the extended (second-order) system (1), (4), the availability of smooth stabilizing laws $v = v_d(x, y, \theta)$ and $\omega = \omega_d(x, y, \theta)$ for the first-order system (1) can be exploited in different ways. We describe in the following two different approaches, based on *backstepping* (see, e.g., [10]) and on the theory of *cascaded systems* (see, e.g., [11]). In the following, it will be assumed that the actual linear and angular velocities of the platform, v and ω, are available for measurements.

A. Backstepping

This technique provides a constructive systematic method to devise globally stabilizing control laws for the general class of triangular nonlinear systems of the form

$$
\begin{aligned}
\dot{\xi} &= f(\xi) + g(\xi)\xi_1 \\
\dot{\xi}_1 &= f_1(\xi, \xi_1) + g_1(\xi, \xi_1)\xi_2 \\
\dot{\xi}_2 &= f_2(\xi, \xi_1; \xi_2) + g_2(\xi, \xi_1; \xi_2)\xi_3 \\
&\vdots \\
\dot{\xi}_n &= f_n(\xi, \xi_1, \dots, \xi_n) + g_n(\xi, \xi_1, \dots, \xi_n)u.
\end{aligned} \qquad (12)
$$

This is obtained in the following way. First consider ξ_1 as a virtual input to stabilize the first subsystem. Let ξ_1^d be the

---

[1]The saturation in ω_f is formally necessary to exclude a possible divergence when (x, y) approaches the origin. However, for V_w smooth enough, the platform tends to align with V_w so that ω_f is practically zero at steady state.

control law that solves this first subproblem, and assume that $V_1(\xi)$ is a corresponding Lyapunov function. Next, define z_1 to be the difference between ξ_1 and its desired value ξ_1^d, and consider ξ_2 as a virtual input to stabilize the subsystem with states ξ and z_1. In order to find a desired behavior for ξ_2, a Lyapunov function candidate is obtained by simply augmenting $V_1(\xi)$ with a quadratic term in z_1, i.e.,

$$V_2(\xi, \xi_1) = V_1(\xi) + \frac{1}{2} z_1^T z_1, \qquad (13)$$

and then ξ_2 is chosen so as to make $\dot{V}_2 < 0$. Proceeding step by step along these lines, one finally arrives at a control law for u.

For system (1) with the dynamic extension (4), the structure (12) is recovered by setting

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \xi_1 = \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad \xi_2 = u = \begin{bmatrix} a \\ \eta \end{bmatrix}, \quad (14)$$

$$f(\xi) = f_1(\xi, \xi_1) = 0, \quad g(\xi) = A(x, y, \theta), \quad g_1(\xi, \xi_1) = I_2,$$

where I_2 is the 2×2 identity matrix and A is defined in (1). The first step of the backstepping procedure is completed by choosing, e.g., ξ_1^d as the stabilizing control law (11). Correspondingly, we can select the Lyapunov function $V_1(\xi)$ along the lines given in [8]. At the second step of the procedure, we write the Lyapunov function V_2 as in (13), with $z_1 = \xi_1 - \xi_1^d$. Along the system trajectories it holds

$$\dot{V}_2 = \frac{\partial V_1}{\partial \xi} A(\xi) \xi_1 + (\xi_1 - \xi_1^d)^T (\xi_2 - \dot{\xi}_1^d). \qquad (15)$$

From (15), it is clear that a control law making the overall system (1), (4) asymptotically stable is

$$u = \xi_2 = \dot{\xi}_1^d - K(\xi_1 - \xi_1^d) - A^T(\xi) \frac{\partial V_1}{\partial \xi}^T, \qquad (16)$$

for any positive definite 2×2 matrix K. Correspondingly, it is

$$\dot{V}_2 = \dot{V}_1^d - (\xi_1 - \xi_1^d)^T K (\xi_1 - \xi_1^d) < 0,$$

where $\dot{V}_1^d < 0$ being the derivative of the Lyapunov function $V_1(\xi)$ along the trajectories of subsystem (1), when $\xi_1 = \xi_1^d$. Note that the convergence of ξ_1 to ξ_1^d is also implied by the control law (16). However, the computation of $\frac{\partial V_1}{\partial \xi}$ is required for the implementation of this control law.

B. Cascaded approach

An alternative and more convenient method for deriving a second-order feedback law based on the available first-order control of Sect. III relies on a result about the stability of cascaded systems (see [11] for details). Under suitable conditions, the stability of the system

$$\begin{aligned} \dot{\zeta}_1 &= f_1(t, \zeta_1) + g_1(t, \zeta_1, \zeta_2)\zeta_2 \\ \dot{\zeta}_2 &= f_2(t, \zeta_2). \end{aligned} \qquad (17)$$

can be concluded from the stability of the two subsystems $\dot{\zeta}_1 = f_1(t, \zeta_1)$ and $\dot{\zeta}_2 = f_2(t, \zeta_2)$.

In order to put system (1), (4) in the form (17), it is sufficient to set

$$\zeta_1 = \xi = \begin{bmatrix} x & y & \theta \end{bmatrix}^T, \qquad \zeta_2 = \xi_1 - \xi_1^d,$$

so that the system equations become

$$\dot{\zeta}_1 = A(\zeta_1)\xi_1^d(\zeta_1) + A(\zeta_1)\zeta_2, \qquad (18)$$

$$\dot{\zeta}_2 = u(t) - \frac{d\xi_1^d(t)}{dt} = f_2(t), \qquad (19)$$

being in this case f_2 independent from ζ_2. From the results of Sect. III, we already know that the "downstream" system (18) is asymptotically stable for $\zeta_2 = 0$, i.e., for $\xi_1 = \xi_1^d$. In order to stabilize the overall cascaded system, it is sufficient to stabilize the "upstream" system (19) to the origin, i.e., to bring ζ_2 to zero. This can be easily obtained by the control law

$$u = \dot{\xi}_1^d - K\zeta_2, \qquad (20)$$

for any positive definite matrix K. Compared with the backstepping law (16), the control law (20) is certainly simpler (it does not require the knowledge of V_1) and has been thus selected for implementation.

Note that acceleration control law (20), as well as (16), requires the differentiability of the velocity control law ξ_1^d. Indeed, eqs. (7–8) are not differentiable at configurations where the argument of the sign function is zero. This problem, however, can be simply overcome by setting $d(\text{sign}\,z)/dz \equiv 0$, for all z.

When the walker is in motion (i.e., when considering the system equations (3–4)), the velocity-level control law is simply obtained by complementing (7–8) with the feedforward action and the observer as in (11). Note that an analytical expression of $\dot{\xi}_1^d$ can be computed from (7–8) and the model (3–4). In this case, $\dot{\xi}_1^d$ is evaluated by assuming that the walker velocity V_w is locally constant.

V. EFFECTS OF PLATFORM MOTION ON THE WALKER

Due to the platform motion, the 'virtual world' frame attached to the walker is in general non-inertial. In particular, even when the intentional velocity of the walker is constant, she/he will feel 'apparent' accelerations (and thus forces) due to the rotation and/or not uniform translation of the carpet. These accelerations must be evaluated in order to verify that they do not exceed the limits of physiological comfort. One major advantage of having moved the control action to the acceleration level is that these computation can be reliably performed in analytical form, since platform acceleration commands are available and it is not necessary to resort to numerical differentiation. In particular, when walking at constant velocity ${}^wV_w$ in the non-inertial virtual world, the total apparent acceleration felt by the user equals the opposite of her/his absolute acceleration. This is computed by analytic differentiation of the first two equations in (1). The apparent acceleration can be decomposed into three different components depending, respectively, on the linear and angular accelerations of the reference frame (*inertial acceleration*),

on the square of the frame angular velocity (*centrifugal acceleration*), and on the coupling between the frame angular velocity and the walker intentional linear velocity (*Coriolis acceleration*). These components should be expressed in the frame (X_w, Y_w, Z_w) attached to the walker, in order to evaluate their physiological effects on the user. The results of the described computations are reported below, where $\text{Rot}(\beta)$ is the 3×3 orthonormal matrix associated to a rotation of a generic angle β around the $Z = Z_w$ axis:

- Inertial component

$$^w a_{in} = \text{Rot}(-\theta_w)\left(\text{Rot}(\theta) \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} + \eta \begin{bmatrix} -y \\ x \\ 0 \end{bmatrix} \right);$$

- Centrifugal component

$$^w a_{cen} = \omega^2 \text{Rot}(-\theta_w) \begin{bmatrix} x & y & 0 \end{bmatrix}^T;$$

- Coriolis component

$$^w a_{Cor} = 2 \begin{bmatrix} 0 & 0 & \omega \end{bmatrix}^T \times {}^w V_w,$$

where the symbol $\times$ denotes the vectorial product.

VI. SIMULATION RESULTS

In order to evaluate the performance of the proposed acceleration control scheme (20), we report here the results for the *virtual* square path with 3 m sides shown in Fig. 4. The walker starts at rest from the initial absolute position $(0, 1)$ m (denoted by a triangle) and moves along each edge with a trapezoidal velocity profile, having symmetric acceleration/deceleration phases with 2.4 m/s^2 for 0.5 s each and a cruise velocity of 1.2 m/s kept for 2 s. At each corner, the walker stops and turns ccw with an angular speed of $\pi/2$ rad/s. Thus, the total trajectory lasts 16 s. Without motion control of the platform, the walker would exit from the platform circular boundary (set at a radius of 2.5 m). In Fig. 5, the absolute motion of the walker (as seen by the overlooking camera) is shown. The absolute orientation θ_w of the walker is displayed by a segment. Initially, we have $\theta = 0$ and $\theta_w = \pi/2$. The gain matrix in (20) is $K = \text{diag}(20, 20)$, while the 'reference' behavior ξ_1^d for the linear and angular carpet velocities is given by the smooth feedback law (7–8) used in scheme (11). The associated control parameters are $k = 1$, $k_w = 10$, and $|\omega_f| \leq 2$ rad/s (to be used for defining the $\text{sat}(\cdot)$ function in (11)). The corresponding linear and angular acceleration commands are shown in Fig. 6. Furthermore, in Fig. 7, the resulting linear and angular velocities of the carpet (blue, solid) are compared with the velocity commands (red, dashed) generated by the first-order velocity controller (7–8) and (11) under the same walker path. This comparison confirms that the second-order controller achieves the same performance of the first-order law, while allowing a direct monitoring of system accelerations. In particular, thanks to the combined feedback and feedforward actions, the moving walker is rapidly brought close to the platform center and then kept there. As in the case of velocity control (see [8]), the platform lags behind any turn performed on place by the walker at the
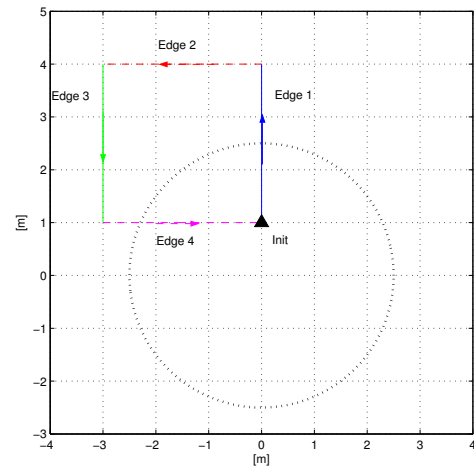

Fig. 4. Virtual square path. The walker moves counterclockwise starting from the Init point (a dotted circle represents the platform boundary)
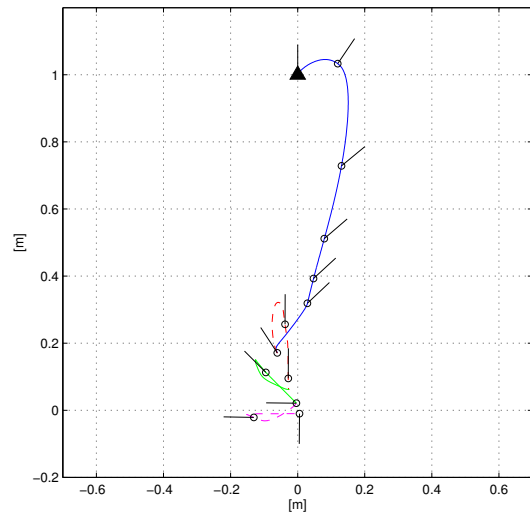

Fig. 5. Virtual square path. Walker absolute locomotion under the second-order platform controller (20)

path corners. In fact, there is no feedforward action triggered by a walker's angular motion without linear displacement. Finally, Figure 8 displays the inertial, centrifugal and Coriolis components of the apparent acceleration felt by the user in the X_w and Y_w directions due to the carpet accelerations of Fig. 6, and computed as in Sect. V. The overall motion of the platform/walker system is also shown in the accompanying video clip. Videos of this and other motion tasks are available at `http://www.dis.uniroma1.it/~labrob/research /CW.html`.

VII. CONCLUSIONS

Using a second-order kinematic model of the *CyberCarpet* nonholonomic platform, we developed acceleration feedback laws that are able to safely keep the walking user close to the platform center, using only measures of the current walker's position and of the state of the turntable (angular position, linear and angular velocities). The presented acceleration
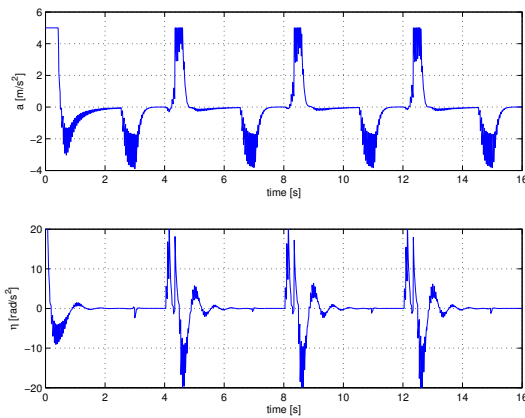
Fig. 6. Linear (top) and angular (bottom) acceleration commands
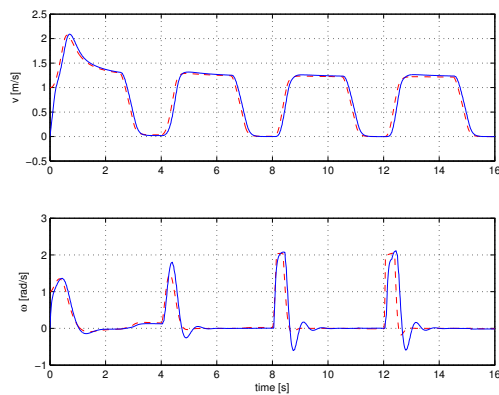

Fig. 7. Carpet linear (top) and angular (bottom) velocities (blue, solid) corresponding to the acceleration commands of Fig. 6, compared with the velocity commands (red, dashed) generated by the first-order feedback controller (7–8) and (11) under the same walker path of Fig. 4
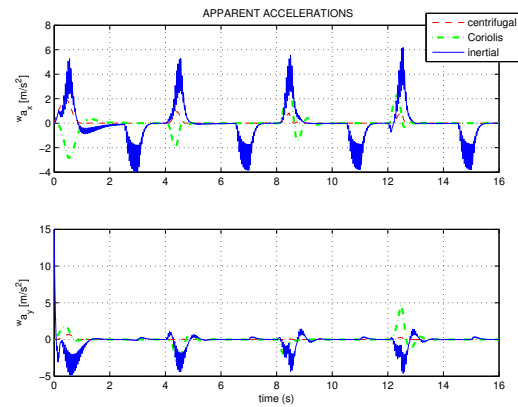

Fig. 8. Inertial, centrifugal and Coriolis components of the apparent acceleration felt by the user in the X_w (top) and Y_w (bottom) directions, due to the carpet accelerations of Fig. 6

References

[1] *CyberWalk*, "EU STREP Project FP6–511092, http://www.cyberwalk-project.org," 2005.
[2] H. Iwata, "Locomotion interface for virtual environments," in *Proc. 9th Int. Symp. on Robotics Research*, pp. 275–282, 2000.
[3] J. M. Hollerbach, "Locomotion interfaces," in *Handbook of Virtual Environments Technology* (K. M. Stanney, ed.), pp. 239–254, 2002.
[4] R. Darken, W. Cockayne, and D. Carmein, "The Omnidirectional Treadmill: A locomotion device for virtual worlds," in *Proc. Symp. User Interface Software and Technology*, pp. 213–221, 1997.
[5] H. Iwata, "The Torus Treadmill: Realizing locomotion in VEs," *IEEE Computer Graphics and Applications*, vol. 9, pp. 30–35, 1999.
[6] A. Nagamori, K. Wakabayashi, and M. Ito, "The Ball Array Treadmill: A locomotion interface for virtual worlds," in *Work. on New Directions in 3D User Interfaces (at VR 2005)*, (Bonn, D), 2005.
[7] A. De Luca, R. Mattone, and P. Robuffo Giordano, "The motion control problem for the *CyberCarpet*," in *Proc. 2006 IEEE Int. Conf. on Robotic and Automation*, (Orlando, FL), 2006.
[8] A. De Luca, R. Mattone, and P. Robuffo Giordano, "Feedback/feedforward schemes for motion control of the *CyberCarpet*," in *Proc. 2006 IEEE Symp. on Robot Control*, (Bologna, I), 2006.
[9] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics & Automation Mag.*, vol. 2, pp. 27–35, 1995.
[10] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control, John Wiley and Sons, 1995.
[11] E. Panteley and A. Loría, "On global uniform asymptotic stability of nonlinear time-varying systems in cascade," *Systems and Control Letters*, vol. 33, no. 2, pp. 131–138, 1998.

control laws build upon the first-order feedback/feedforward schemes introduced in [8], that are able to deal with control singularities of the platform kinematics and to compensate for the walker voluntary motion, through the addition of a feedforward term based on an observer of the walker's linear velocity. In particular, the design of the second-order control law relies on the stability of the first-order scheme and can be based either on the technique of *backstepping* [10], or on the theory of *cascaded systems* [11]. The latter is more convenient for implementation. Simulation results, and the accompanying video clip, demonstrate the obtained performance.

The availability of the platform accelerations allows also the analytical computation of the apparent accelerations felt by the user during the motion. The consideration of explicit bounds on these accelerations, and the experimental validation of the second-order scheme on a prototype of the platform will be the subjects of future work.

Acknowledgments