

FEEDBACK/FEEDFORWARD SCHEMES FOR MOTION CONTROL OF THE CYBERCARPET

A. De Luca, R. Mattone, P. Robuffo Giordano¹

*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 possibilities to 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. Simple but global kinematic control schemes are presented, addressing in particular the handling of singularity issues. The feedback stabilizing part, which is based only on the user's pose information, is complemented by a feedforward term derived from on a walker's velocity observer. Numerical and graphical simulation results are presented.

Keywords: Nonholonomic platform, nonlinear stabilization, velocity observers

1. INTRODUCTION

Exploration of virtual reality worlds by allowing omni-directional unconstrained locomotion possibilities for a walking user is the main objective of the European research project (*CyberWalk*, 2005).

Different locomotion interfaces already exist that allow walking in virtual environments (see (Iwata, 2000) and (Hollerbach, 2002)). However, they usually constrain the feet or the legs of the user. For unconstrained planar walking, the Omnidirectional Treadmill has been proposed in (Darken *et al.*, 1997) using two perpendicular belts and a large number of rollers, while a torus-shaped belt arrangement is implemented in the Torus Treadmill (Iwata, 1999).

A different principle is used in (Nagamori *et al.*, 2005), 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 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 will be allowed to execute slow or fast locomotion in a natural way and in any planar (possibly infinite) direction. The platform controller will counteract her/his motion and pull the walker toward the center of the *CyberCarpet*, taking into account physiologically acceptable velocity/accelerations bounds. The body pose on the carpet is acquired through a markerless visual tracking system and drives 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, including one preliminary design of a circular *CyberCarpet*, is shown in Fig. 1.

¹ Work supported by EU Project FP6-511092 *CyberWalk*

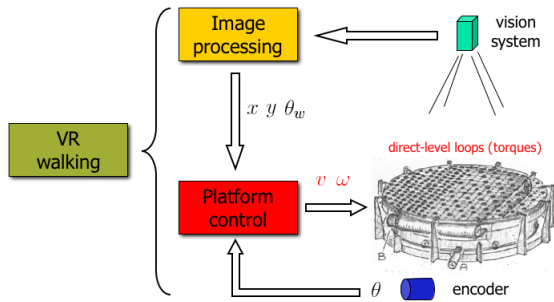


Fig. 1. Control system architecture of the *CyberCarpet*, with a drawing of the preliminary platform design (courtesy of Max Planck Institute for Biological Cybernetics; German Patent filed in 2005)

Previous works on locomotion interfaces have paid little attention to control issues and algorithms, relying on heuristic schemes with no stability or performance analysis or restricting to low and piece-wise constant walker velocity with few directional changes. In our first work on the subject (De Luca *et al.*, 2006), we developed a first-order kinematic model of the *CyberCarpet* platform and recognized a duality with the control problem for nonholonomic wheeled robots. We proposed then a velocity control scheme based on input-output decoupling and linearization, yielding a quite satisfactory overall performance. While transient recovery could be easily specified on the linear side of the problem (including an integral action to compensate at steady state for a constant walker velocity), the control law had some annoying singularities making its validity not global.

In this paper, after recalling the kinematic modeling of the platform (Sect. 2), we present two modified feedback control schemes handling the previous singularities, and include also a feedforward term based on an observer of the (unavailable) instantaneous velocity of the walker (Sect. 3). The performance is evaluated by simulation in Sect. 4. Conclusions and future work within the *CyberWalk* project are discussed in the final section.

2. KINEMATIC MODELING

A first-order kinematic model of the *CyberCarpet* has been derived in (De Luca *et al.*, 2006). 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.

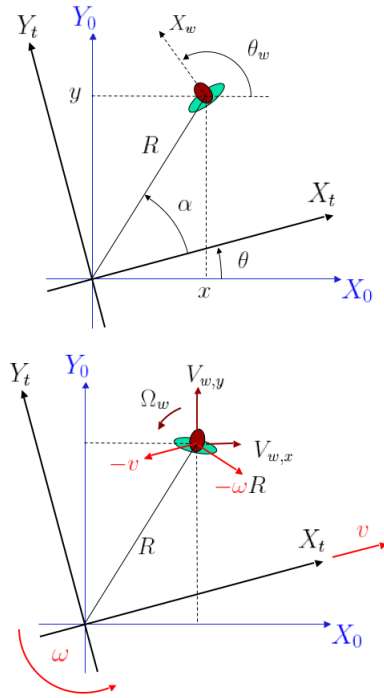


Fig. 2. Frames and variables definition: walker and platform still (top) and in motion (bottom)

With reference to Fig. 2, 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

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

being v and ω the commanded linear and angular velocity of the *CyberCarpet*.

From eq. (1), it is clear that a holonomic constraint exists, i.e., $\theta + \theta_w = \text{const}$, so that only one of these two variables can be independently controlled (this is not a limitation for the considered control task). In the resulting three-dimensional configuration space, the system is fully controllable being subject to a completely nonholonomic differential constraint (De Luca *et al.*, 2006).

For later use, note that the two Cartesian coordinates (x, y) may be replaced by suitable polar coordinates (see also (Aicardi *et al.*, 1995)), with $R = \sqrt{x^2 + y^2}$ being the walker's distance from the center and angle $\alpha = \text{atan2}(y, x) - \theta$ locating the position of the walker in the rotated frame (X_T, Y_T) . One obtains

$$\begin{aligned}\dot{R} &= -v \cos \alpha \\ \dot{\alpha} &= v \frac{\sin \alpha}{R} - 2\omega\end{aligned}\quad (2)$$

When the walker is in motion, the kinematic model becomes

$$\begin{aligned}\dot{x} &= -v \cos \theta + y\omega + V_{w,x} \\ \dot{y} &= -v \sin \theta - x\omega + V_{w,y} \\ \dot{\theta} &= \omega \\ \dot{\theta}_w &= -\omega + \Omega_w,\end{aligned}\quad (3)$$

with absolute linear and angular walker velocities $V_w = (V_{w,x}, V_{w,y})$ and Ω_w , respectively (see bottom of Fig. 2).

3. CONTROL DESIGN

Formally, the control task requires an output 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 would be sufficient, since the practical objective is just to keep the walker safely within the physical boundaries of the platform. The 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. The control outputs are velocity commands which are taken as references in the direct-level PIDs servoing the two platform actuators for linear (belt) and angular (turntable) motion. Although linear and angular velocities (and accelerations) felt by the walker should be kept within physiological limits, we do not take these constraints explicitly into account in this study, which serves as a basis for further control developments.

In (De Luca *et al.*, 2006), we presented a simple control design based on input-output feedback linearization (leading to two decoupled single input-output integrators as intermediate step). When the walker stands still, this control law achieves exponential stabilization of the walker position to the origin, with arbitrary and independent rates for the x and y components. For the convenient case of equal convergence rates $k > 0$, the resulting control law is given by

$$v = \frac{k(x^2 + y^2)}{x \cos \theta + y \sin \theta} = \frac{kR}{\cos \alpha}\quad (4)$$

and

$$\omega = \frac{k(y \cos \theta - x \sin \theta)}{x \cos \theta + y \sin \theta} = k \tan \alpha,\quad (5)$$

which are expressed in terms of either the Cartesian errors x and y or the radial and angular errors R and α .

The Cartesian expression of this decoupling control law has the singularity $x \cos \theta + y \sin \theta = R \cos \alpha = 0$ (where the determinant of the decoupling matrix is zero), i.e., either when the walker stands on the Y_T axis or at the origin. On the other hand, the expression in polar coordinates clearly shows the control singularity at $\alpha = \pm \frac{\pi}{2}$, while the singularity at the origin of the (x, y) space is hidden by the fact that the angle α is not defined there. However, a step of $\pm \pi$ rad in the value of α is experienced when crossing the origin, which makes the control input ω change sign. As a consequence, even measurement noise on the x and y variables can cause an undesired chattering in the value of ω when the walker approaches the platform center.

In the following we shall introduce modifications that deal with these control singularities, while trying to preserve some convenient characteristics of the decoupling law (4–5). We consider first the case of absence of disturbances, $V_w = 0$ and $\Omega_w = 0$ (walker standing still in the virtual environment), dealing later with the motion of the walker (Sec. 3.3).

3.1 Singularity at $\cos \alpha = 0$

The singularity at $\alpha = \pm \frac{\pi}{2}$ can be eliminated by simply taking

$$v = kR \operatorname{sgn}(\cos \alpha),\quad (6)$$

and

$$\omega = k \sin \alpha \operatorname{sgn}(\cos \alpha),\quad (7)$$

with $\operatorname{sgn} x = 1$ for $x \geq 0$ and $\operatorname{sgn} x = -1$ otherwise. Control laws (6–7) are formally obtained by multiplying eqs. (4–5) by $|\cos \alpha|$. The resulting closed-loop input-output dynamics is now

$$\dot{x} = -k |\cos \alpha| x, \quad \dot{y} = -k |\cos \alpha| y,\quad (8)$$

which is not linear nor decoupled anymore, since the angle α depends on both x and y . However, note that from (8) it follows

$$\frac{y(t)}{x(t)} = \frac{\dot{y}(t)}{\dot{x}(t)} = \frac{y_0}{x_0},\quad (9)$$

just as with the decoupling and linearizing law (4–5). Therefore, a user standing still will be driven along the straight line connecting its initial position to the origin ($\theta + \alpha = \beta_0 = \text{const}$).

In order to show that the control law (6–7) asymptotically stabilizes the walker position (x, y) to

the origin, consider the positive definite Lyapunov function

$$\begin{aligned}\mathcal{V}(x, y, \theta) &= \frac{1}{2}(x^2 + y^2 + \sin^2(\beta_0 - \theta)) \\ &= \frac{1}{2}(x^2 + y^2 + \sin^2 \alpha),\end{aligned}$$

for which it is $\dot{\mathcal{V}} = 0$ iff

$$(x, y, \theta) \in \mathcal{S} = \{(0, 0, \bar{\theta}), \sin(\beta_0 - \bar{\theta}) = 0\}.$$

Taking into account eqs. (1–2), the derivative of \mathcal{V} along the system trajectories is given by

$$\begin{aligned}\dot{\mathcal{V}} &= x\dot{x} + y\dot{y} + \sin \alpha \cos \alpha \dot{\alpha} \\ &= -k |\cos \alpha| (x^2 + y^2 + \sin^2 \alpha) \leq 0,\end{aligned}$$

so that $\dot{\mathcal{V}} = 0$ for $(x, y, \theta) \in \mathcal{S}$ as well as for $\cos \alpha = 0$. However, the latter does not correspond to closed-loop system equilibria, since $\omega = \pm k$ in these configurations. Therefore, \mathcal{S} is the largest invariant set for the system, and (x, y) converges to the origin by virtue of LaSalle’s theorem.

3.2 Singularity at $R = 0$

The control law (6–7) is clearly not continuous at the origin, due to the discontinuity of the angle α which is not defined at $x = y = 0$. This causes a chattering of the control input ω when the walker is in a small region around the platform center². In order to avoid this problem, we propose next two different strategies.

3.2.1. Dead zone around the origin A simple way to overcome input chattering when the walker approaches the platform center is to introduce a small dead zone for the input commands v and ω around the origin, i.e., setting $v = \omega = 0$ for $R < \bar{R}$. At the cost of a small (namely, of amplitude \bar{R}) error in the final configuration, which is consistent with the task specification, this strategy is sufficient to solve the problem in the absence of walker’s motion: the platform stops as soon as the walker is brought inside the dead zone. However, when the walker is in locomotion, the chattering of the input commands may appear again at the border of the dead zone. In order to at least reduce the frequency of this chattering, some *hysteresis* can be introduced at the border of the dead zone by defining two different thresholds, \bar{R}_{off} and \bar{R}_{on} , for triggering the switch-off and switch-on of the control law (6–7). This heuristic approach usually shows a satisfactory performance both in terms

² Although a term $\text{sgn}(\cos \alpha)$ appears also in the expression of v , the chattering phenomenon for this control input is overcome by the presence of the factor R vanishing at the origin.

of input commands behavior and of walker’s executed path.

3.2.2. A smooth nonsingular control law A more elegant solution to the chattering problem is obtained by replacing the feedback control laws (6–7) with

$$\begin{aligned}v &= kR^2 \text{sgn}(\cos \alpha) \\ &= k(x^2 + y^2) \text{sgn}(x \cos \theta + y \sin \theta),\end{aligned}\quad (10)$$

and

$$\begin{aligned}\omega &= kR \sin \alpha \text{sgn}(\cos \alpha) \\ &= k(y \cos \theta - x \sin \theta) \text{sgn}(x \cos \theta + y \sin \theta),\end{aligned}\quad (11)$$

which are formally obtained by multiplying eqs. (6–7) by the radial distance R . The control laws (10–11) are defined at any system configuration. Furthermore, the angular velocity command ω converges to zero as (x, y) approaches the origin, so that no chattering problem exists in this case. Under the feedback laws (10–11) and in the absence of walker locomotion, the closed-loop input-output dynamics becomes

$$\dot{x} = -kR |\cos \alpha| x, \quad \dot{y} = -kR |\cos \alpha| y, \quad (12)$$

which can be proved to be asymptotically stable at the origin by the same Lyapunov arguments used for laws (6–7). Furthermore, note that eq. (9) still holds for the dynamics (12), i.e., the user is pulled toward the origin along the connecting straight line. Finally, in comparison with the discontinuous control modification of Sect. 3.2.1, the smooth feedback (10–11) does not exhibit any steady-state error. However, the convergence rate of the x and y variables drops quadratically to zero as the walker approaches the origin.

3.3 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 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 (6–7). In particular, it can be shown that, when the user walks indefinitely in the virtual environment along a straight line with constant velocity \bar{V} , a steady-state position at a distance $\bar{R} = \bar{V}/k$ from the origin will be reached, as in the case of the original decoupling and linearizing law in (De Luca *et al.*, 2006). There, using standard results from linear control theory, we added an integral control action in order to completely eliminate this steady-state error. However, a poor dynamic performance results

due to the typical output overshooting associated to the presence of an integral control term.

We propose here a different approach to deal with walker's locomotion. In particular, based on an estimate \tilde{V}_w of the walker linear velocity V_w , we shall include a suitable feedforward term in eqs. (6–7) or in eqs.(10–11) as follows

$$\begin{aligned} v_c &= v + v_f = v + [\cos \theta \ \sin \theta] \tilde{V}_w \\ \omega_c &= \omega + \omega_f \\ &= \omega + \text{sat} \left(\frac{1}{R} [-\sin \theta \ \cos \theta] \tilde{V}_w \right). \end{aligned} \quad (13)$$

Here, $\text{sat}(\cdot)$ is the 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 eqs. (13) 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 direction³.

In order to get an accurate estimate \tilde{V}_w of the walker velocity, consider the two scalar dynamic systems (viz. velocity observers)

$$\begin{aligned} \dot{\xi}_x &= -v \cos \theta + y\omega + k_w(x - \xi_x) \\ \tilde{V}_{w,x} &= k_w(x - \xi_x), \end{aligned} \quad (14)$$

and

$$\begin{aligned} \dot{\xi}_y &= -v \sin \theta - x\omega + k_w(y - \xi_y) \\ \tilde{V}_{w,y} &= k_w(y - \xi_y), \end{aligned} \quad (15)$$

where $k_w > 0$. From eqs. (14–15) and (3), it follows

$$\begin{aligned} \dot{\tilde{V}}_{w,x} &= k_w (V_{w,x} - \tilde{V}_{w,x}) \\ \dot{\tilde{V}}_{w,y} &= k_w (V_{w,y} - \tilde{V}_{w,y}), \end{aligned}$$

i.e., the estimates are low-pass filtered versions of absolute Cartesian components of V_w . In particular, for k_w large enough, they accurately reproduce the two components $V_{w,x}$ and $V_{w,y}$ of the absolute walker velocity.

Note that, even after this feedforward compensation, the system is still affected by a residual disturbance

$$V_w - \tilde{V}_w = \frac{s}{s + k_w} V_w.$$

Therefore, walker's constant velocities are fully compensated at steady state for any positive

³ 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.

k_w , while for walker's ramp-wise velocities (constant accelerations) the associated steady-state error can be made in principle arbitrarily small by increasing k_w —an *astatic* velocity disturbance behavior is recovered by using the feedback/feedforward control scheme (13–15).

4. SIMULATION RESULTS

We present some selected results obtained with the motion control laws discussed above. In all case studies, the walker starts at rest from the initial absolute position $(0, 1)$ m —one that would immediately lead to control problems for the law in (De Luca *et al.*, 2006).

Figures 3–6 refer to the walker moving indefinitely along a straight line directed along the Y_T -axis in the *virtual space* (i.e., for a fixed platform), with a constant velocity of 1 m/s.

In Fig. 3, the actual motion of the walker in the *absolute space* 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 platform is controlled by the combined feedback/feedforward scheme (13), wherein the feedback law (6–7) is used together with a dead zone/hysteresis as described in Sect. 3.2.1. The associated control parameters are: $k = 1$, $k_w = 10$, $\bar{R}_{\text{off}} = 0.05$ m, $\bar{R}_{\text{on}} = 0.1$ m, and $|\omega_f| \leq 2$ rad/s. We note that there is a small error at steady state, just within the distance \bar{R}_{off} , where the input commands are simply $v_c = v_f$ and $\omega_c = \omega_f$ (the dead zone applies only to the feedback terms). At the end, the platform is oriented in the same direction of the walker's motion, at an angle $\theta = \theta_w = \pi/4$. The linear and angular velocity commands shown in Fig. 4 confirm that the feedforward terms converge to the exact velocity compensation ($v_f = 1$ m/s and $\omega_f = 0$).

The same task is performed again with the feedback/feedforward scheme (13), but using the feedback law (10–11) and the same relevant control parameters as before. A zero final position error is obtained (Fig. 5). The overall control behavior in Fig. 6 is clearly smoother, but transients of the input commands v and ω are somewhat longer.

As a more complex motion, we report the results for the *virtual* square path with 3 m sides shown in Fig. 7. The walker starts at rest and moves along each edge with a trapezoidal velocity profile, having symmetric acceleration/deceleration phases with 2.4 m/s² for 0.5 s each and a cruise velocity of 1.2 m/s kept for 2 s. At each reached 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

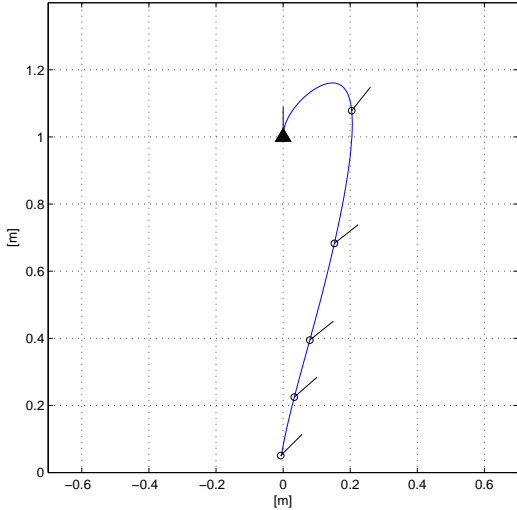


Fig. 3. Virtual straight line: Walker absolute locomotion under the platform controller (13), using the feedback law (6–7) and a dead zone with hysteresis around the origin

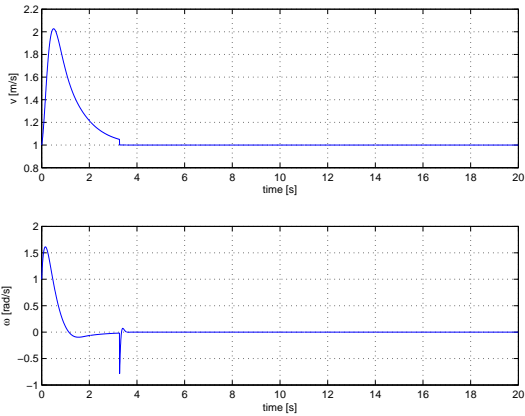


Fig. 4. Linear and angular velocities for Fig. 3

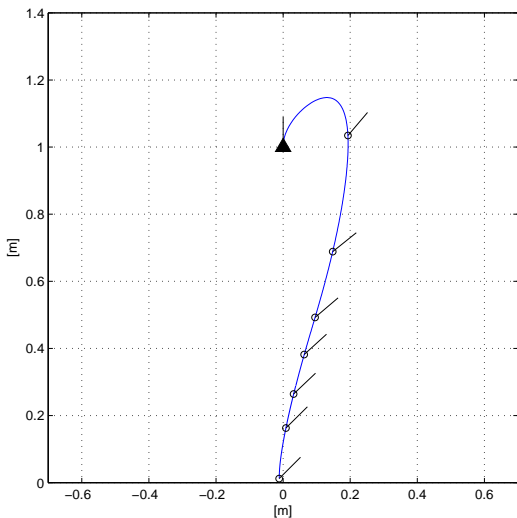


Fig. 5. Virtual straight line: Walker absolute locomotion under the platform controller (13) and using the smooth feedback law (10–11)

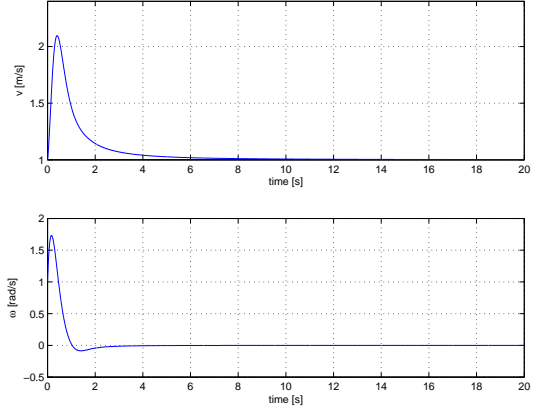


Fig. 6. Linear and angular velocities for Fig. 5

platform, the walker would exit from the platform circular boundary of radius 2.5 m.

We report only the results for the smooth feedback law (10–11) used in scheme (13), see Figs. 8–9. The control parameters are chosen as before. Thanks to the combined feedback and feedforward actions, the walker is rapidly brought close to the platform center and then kept there. The linear control input is smooth and, after an initial transient, does never exceed the walker’s voluntary speed. On the other hand, saturation on the angular feedforward term ($|\omega_f| \leq 2$ rad/s) becomes relevant when the walker is close to the origin and takes a sharp turn (starting with the corner after edge 2). Note that the platform lags behind any turn performed on place by the walker, since there is no feedforward action triggered by a walker’s angular motion without linear displacement. Finally, it can be recognized that a periodic behavior is reached starting with the third edge of the square path.

In order to fully appreciate the overall motion of the platform/walker system, a 3D graphical simulation environment has also been developed, using Simulink and Visual Nastran. Videos of the presented motion tasks are available at <http://www.dis.uniroma1.it/~labrob/research/CW.html>.

5. CONCLUSIONS

Using a kinematic model of the *CyberCarpet* non-holonomic platform, we developed nonlinear 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 the angular position of the turntable. The presented control designs build upon a previous input-output decoupling and linearization law of the cartesian motion of the walker (De Luca *et al.*, 2006), eliminating singularities while preserving excellent convergence characteristics. In absence of direct measurements of the walker voluntary velocity, the ad-

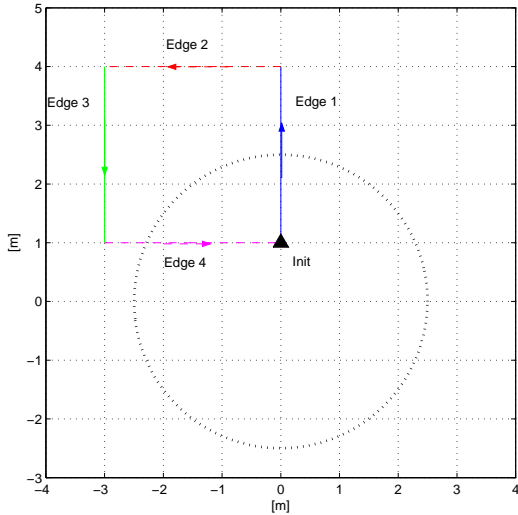


Fig. 7. Virtual square path: Walker moves counterclockwise starting from Init point (a dotted circle represents the platform boundary)

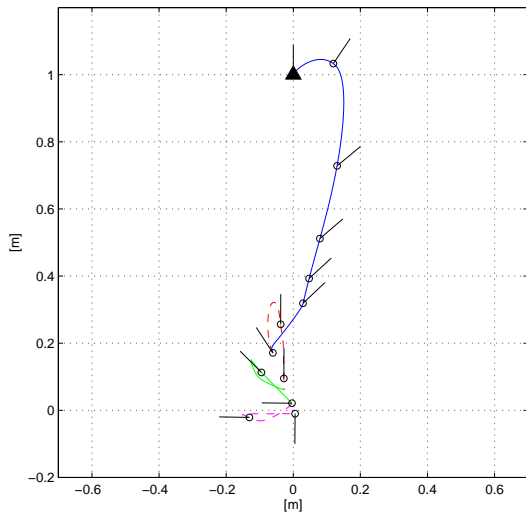


Fig. 8. Virtual square path: Walker absolute locomotion under the platform controller (13) and using the smooth feedback law (10–11)

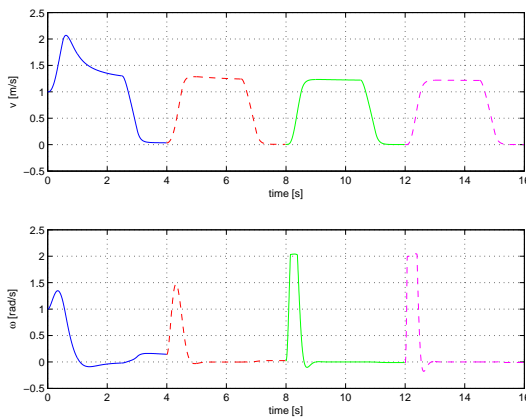


Fig. 9. Linear and angular velocities for Fig. 8

dition of a feedforward term based on an observer of the walker's linear velocity has proved beneficial in terms of control performance.

Future work includes the consideration of explicit bounds on the actuated inputs and of delays and noise in the sensing system, the addition of feedback/feedforward terms depending on the voluntary angular motion of the walker, as well as the derivation of acceleration control laws using a second-order kinematic model of the combined walker/platform system.

REFERENCES

- Aicardi, M., G. Casalino, A. Bicchi and A. Balestrino (1995). Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics & Automation Mag.* **2**, 27–35.
- CyberWalk* (2005). EU STREP Project FP6–511092, <http://www.cyberwalk-project.org>.
- Darken, R., W. Cockayne and D. Carmein (1997). The omnidirectional treadmill: A locomotion device for virtual worlds. In: *Proc. Symp. User Interface Software and Technology*. pp. 213–221.
- De Luca, A., R. Mattone and P. Robuffo Giordano (2006). The motion control problem for the *cybercarpet*. In: *Proc. 2006 IEEE Int. Conf. on Robotic and Automation*. Orlando, FL.
- Hollerbach, J. M. (2002). Locomotion interfaces. In: *Handbook of Virtual Environments Technology* (K. M. Stanney, Ed.). pp. 239–254.
- Iwata, H. (1999). The torus treadmill: Realizing locomotion in ves. *IEEE Computer Graphics and Applications* **9**, 30–35.
- Iwata, H. (2000). Locomotion interface for virtual environments. In: *Proc. 9th Int. Symp. on Robotics Research*. pp. 275–282.
- Nagamori, A., K. Wakabayashi and M. Ito (2005). The ball array treadmill: A locomotion interface for virtual worlds. In: *Work. on New Directions in 3D User Interfaces (at VR 2005)*. Bonn, D.