

The Motion Control Problem for the CyberCarpet

Alessandro De Luca Raffaella Mattone Paolo 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—Exploration of virtual worlds with unconstrained locomotion possibilities for the user is the main objective of the European research project *CyberWalk*. This should be achieved through the use of an actuated platform (the *CyberCarpet*) that compensates for the walker's locomotion in such a way to keep her/him close to the platform center. This paper presents the control problem for the platform motion, including objectives and constraints, overall control architecture, and kinematic modeling. Since the platform has only two actuating devices (linear and angular), the control problem is similar to that of output regulation for nonholonomic wheeled mobile robots in the presence of an unpredictable disturbance due to walker's locomotion. Based on the kinematic model, a velocity control design achieving input-output decoupling and linearization is proposed and its performance is verified by simulations.

I. INTRODUCTION

Physically walking through virtual environments is a challenging goal of recent research in Virtual Reality [1], [2]. The *CyberWalk* European research project [3] plans to advance the state of the art in this area by enabling unconstrained and omni-directional walking in a VR world in a natural way.

For this, a novel concept for a 2D treadmill will be developed, the *CyberCarpet*. A walker will be allowed to execute slow or fast locomotion in any planar direction, or even step over and cross his/her legs, while remaining on the *CyberCarpet* platform. A markerless visual tracking system locates the instantaneous position of the walker on the carpet, providing this information to the platform motion control system. The latter will command two actuation devices linearly moving the belt and rotating the turntable in such a way that the walker is pulled toward its center. The combined walker-platform displacements are used to update the scene of the virtual environment shown on a Head Mounted Display. The motion control problem for the *CyberCarpet* is focused on complying with the omni-directional (and possibly infinite) walk of the user, while keeping velocities and accelerations within physiologically acceptable bounds.

Many different locomotion interfaces exist that allow walking in virtual environments with at least two dimensions (for a 1D system see, e.g., [4]). In the Sarcos Biport, two mechanical arms are attached to the user's feet, applying the proper ground reaction force depending on the movement of the walker [5]. Two parallel mechanisms mounted on a common turntable are used to support the motion of each foot in the Gait Master [6]. Both these devices allow in principle 3D motion (e.g., for climbing stairs or emulating uneven terrains) but constrain

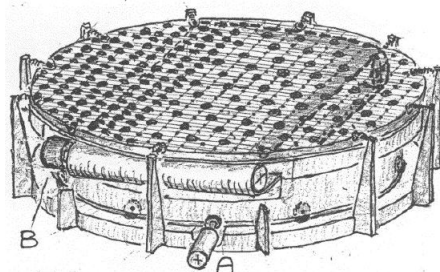


Fig. 1. A drawing of the *CyberCarpet* platform preliminary design (courtesy of Max Planck Institute for Biological Cybernetics; German Patent filed in 2005)

the relative feet motion (legs cannot be crossed). For unconstrained planar walking, the Omnidirectional Treadmill [7] is constituted by two perpendicular belts, with 3400 rollers, that rotate one inside the other, with the upper belt transmitting the lateral motion of the bottom one. A similar arrangement is used in the Torus Treadmill [8], where the locomotion interface is realized by mounting the two belts so as to form a torus-shaped surface. In the CirculaFloor [9], a number of holonomic tiles moves on the floor following the feet motion and compensating for it so that the walking user is kept at the same location. In [10], powered castor wheels with offsets are used to support a platform and counteract the walker motion through friction contacts. Finally, in [11] a conveyor belt and a turntable transmit omni-directional motion to a walker through a ball-array board. The rotating balls are fitted into the array board and are in contact with the belt so that an object on the board moves in the opposite direction of the corresponding point on the belt.

A preliminary design of a circular *CyberCarpet* is shown in Fig. 1. The locomotion principle is similar to that in [11]. However, the scale and loading capability of the *CyberCarpet* will eventually be larger, with associated technological issues in the hardware, while the walker localization sensor is different (visual instead of magnetic). In addition, the whole virtual reality interface is absent in [11].

More in general, the above works on locomotion interfaces pay little attention to control issues and algorithms, and to their performance. For instance, the linear and angular velocity commands in [11] are updated at a slow rate assuming a low and piece-wise constant walker velocity and very few directional changes. No control analysis is reported.

In this paper, we focus on the motion control problem for the *CyberCarpet* platform. The control objective and the overall architecture are described in Sect. II, while a kinematic model is derived in Sect. III, where the nonholonomic nature of the platform motion is recognized. A velocity control scheme based on input-output decoupling and linearization is proposed in Sect. IV and evaluated by simulations in Sect. V. Conclusions and other current work within the *CyberWalk* project are discussed in the final section.

II. CONTROL OBJECTIVES AND ARCHITECTURE

The main objective of motion control for the *CyberCarpet* is to keep the walker absolute position within a sufficiently small distance from the center (a target disk) of the platform, despite of the voluntary and unpredictable locomotion of the user. A suitable radius of such a target disk may be around 10% of the actual size of the platform. Similarly, the evaluation of a safe distance from the platform boundaries should be based on the maximum admissible velocity assumed for the walker locomotion and on the actuator capabilities. Although the knowledge of the walker orientation is needed for the correct display of the virtual environment to the user, this information is not relevant for the stated control task. Moreover, in order to achieve a natural and comfortable operation, the linear and angular velocities and accelerations (as well as inertial forces/moments) felt by the walker should be kept limited. These limits will reflect into state-dependent upper bounds on the input commands to the platform.

The actual values of the above design constraint parameters will be the outcome of physiological studies on human subjects within the *CyberWalk* project [3]. The present study does not explicitly take into account these constraints but serves as a basis for further control developments. The consortium is building first a ball-array platform at small scale (1 m size), in order to evaluate the feasibility of the approach.

The overall control system architecture is shown in Fig. 2. A high-level visual tracker provides the absolute location of the walker on the platform. The height of the camera above the platform center is not critical in terms of the needed resolution. These measurements, together with the angular orientation of the turntable, are available for the low-level platform motion control. This control output drives the two actuators of the turntable and of the belt. In principle, either kinematic or direct torque controllers can be used: in the first case, which is the most common in conventional servo-drives, velocity commands are taken as reference for direct-level PIDs. Finally, the visualization task in the VR-space requires processing of the observed absolute motion of the walker so as to filter out the component induced by the controlled motion of the platform.

For model-based control design, suitable kinematic and/or dynamic models should be derived for describing the motion of the walker/platform system. In this paper, we are mainly interested in assessing the kinematic recovery capabilities of the system, i.e., the way a general displacement of the walker can be compensated by suitable platform motions. For this,

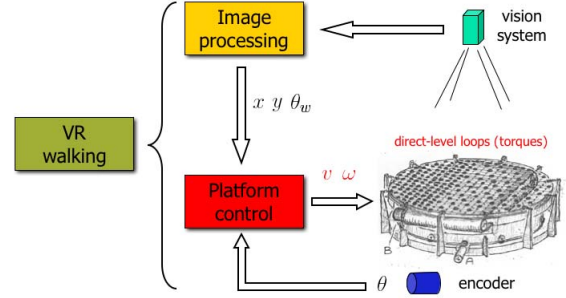


Fig. 2. Control system architecture

a first-order kinematic model will be sufficient. Note that the described feedback control task can be formally classified as an output regulation (of the walker position) in the presence of unknown disturbances (the intended motion of the walker).

III. KINEMATIC MODELING

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, i.e., a forward motion command will move the user backwards, and a clockwise rotation will turn the user counterclockwise. Keeping this in mind, it is possible to derive a first-order kinematic model of the *CyberCarpet* with the help of Fig. 3. Therein, (X_0, Y_0) is the absolute frame (also attached to the fixed overlooking camera) and (X_T, Y_T) is the frame rotated by an angle θ and attached to the treadmill, with the X_T -axis in the direction of the belt (along which linear motion is actuated). Both frames have the origin at the center of the *CyberCarpet*. The walker absolute position and orientation are (x, y) and θ_w , respectively, with $R = \sqrt{x^2 + y^2}$ being his/her distance from the center. The angle $\alpha = \text{atan2}(y, x) - \theta$ locates the position of the walker in the frame (X_T, Y_T) .

As a result, when the walker is standing still, we obtain

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

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

A simple analysis of the kinematic equations (1) shows 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 motion control task. In the resulting three-dimensional configuration space, parametrized for instance by (x, y, θ) , the system is fully controllable being subject to the completely nonholonomic

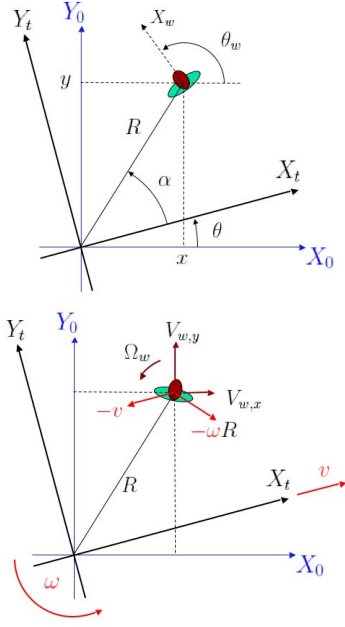


Fig. 3. Frames and variables definition: walker and platform still (top) and in motion (bottom)

differential constraint

$$\begin{bmatrix} \sin \theta & -\cos \theta & -(x \cos \theta + y \sin \theta) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0.$$

In view of this, the motion control problem for the *Cyber-Carpet* is similar to the one for nonholonomic wheeled mobile robots. The analogy of the two problems can be intuitively recognized also by flipping things upside down: the standing user plays the role of the fixed ground, while the nonholonomic platform will act as the moving wheeled robot. This duality is lost when the unconstrained walker starts to move.

In fact, when the walker is in motion, the model becomes

$$\begin{aligned} \dot{x} &= -v \cos \theta + y \omega + V_{w,x} \\ \dot{y} &= -v \sin \theta - x \omega + V_{w,y} \\ \dot{\theta} &= \omega \\ \dot{\theta}_w &= -\omega + \Omega_w, \end{aligned} \quad (2)$$

with absolute linear and angular walker velocities ($V_{w,x}, V_{w,y}$) and Ω_w , respectively (see bottom of Fig. 3). Indeed, these walker's velocities will act as disturbances in the control system and are assumed here unpredictable and not directly measurable.

IV. CONTROL DESIGN

A number of feedback control laws developed for nonholonomic wheeled mobile robots can be modified to address the regulation problem for the *CyberCarpet*, so as to bring the position (and orientation) of a standing user to zero by suitable maneuvers. These techniques include Lyapunov design in polar coordinates [12], time-varying nonlinear control [13], control based on the chained-form transformation [14] or

on system flatness [15], or recursive control with backstepping [16]. Although successful, these techniques lead to somewhat erratic and/or slow transients which are not convenient for the practical case of a walker in motion. Their extension to handle such a type of disturbance has not been considered yet. On the other hand, since only the walker's position has to be asymptotically stabilized to the origin (actually, to an arbitrarily small circle around the origin), we present here a simpler design based on input-output feedback linearization. Since the control law will use only the instantaneous sensor information on the walker position (the system output to be regulated), the closest technique already available for mobile robots is the one based on artificial potentials [17].

Consider first the case of no disturbances, i.e., $V_w = 0$ and $\Omega_w = 0$ (walker standing still in the virtual environment) and define the controlled output as $z = [x \ y]^T$. Differentiating z in time and using eq. (1) gives

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\cos \theta & y \\ -\sin \theta & -x \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = A(x, y, \theta) \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

Assuming $\det A = x \cos \theta + y \sin \theta \neq 0$, we can set

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = A^{-1}(x, y, \theta) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (3)$$

where v_1 and v_2 are auxiliary velocity inputs to be defined. The resulting closed-loop input-output behavior is constituted by simple integrators

$$\dot{z}_1 = \dot{x} = v_1, \quad \dot{z}_2 = \dot{y} = v_2,$$

i.e., it is decoupled and linearized by the feedback law (3). The control design can be completed by the proportional laws

$$v_1 = -k_1 x, \quad v_2 = -k_2 y, \quad (4)$$

with positive gains k_i ($i = 1, 2$), thus exponentially stabilizing the walker's position to the origin.

The above derivations hold outside the singularities of matrix A , i.e., whenever $x \cos \theta + y \sin \theta = R \cos \alpha \neq 0$ (see Fig. 3). For the purpose of analysis, a more convenient expression for v and ω can be found by choosing the gains $k_1 = k_2 = k > 0$ and replacing eq. (4) into (3). This yields

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

and

$$\omega = \frac{k(y \cos \theta - x \sin \theta)}{x \cos \theta + y \sin \theta} = \frac{kR \cos(\frac{\pi}{2} - \alpha)}{R \cos \alpha} = k \tan \alpha. \quad (6)$$

A limit study for R going to zero and $\cos \alpha \neq 0$ shows that v and ω are continuous, with v vanishing and ω remaining bounded. Moreover, the control singularity at $\alpha = \pm \frac{\pi}{2}$ is relevant only at the initial instant if the walker is located on the Y_t -axis (see Fig. 3), in which case it can be easily managed by a simple heuristics applied within a small zone including the Y_t -axis. For all other initial conditions, the angular control law (6) will automatically drive the walker away from this singularity.

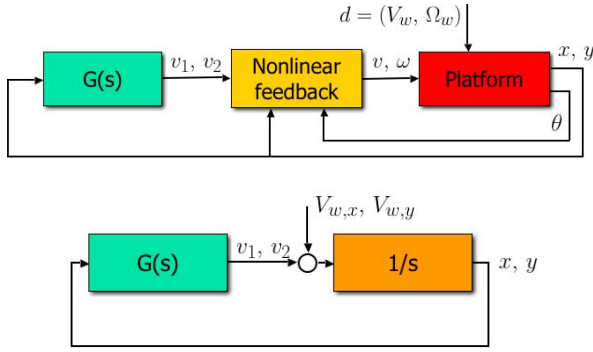


Fig. 4. The nonlinear control scheme (top) and its equivalent input-output linear representation (bottom)

Another interesting property of the designed controller arises from the linear and decoupled behavior of the closed-loop system. With the walker standing in an initial position (x_0, y_0) , the time evolution of his/her position will be forced by eq. (4) to be $x(t) = e^{-k_1 t} x_0$ and $y(t) = e^{-k_2 t} y_0$. With $k_1 = k_2$, it will be

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

so that the user will be pulled toward the origin along the connecting straight line. This will be evident in the numerical simulations of Sect. V.

The overall block diagram of the above nonlinear control scheme is depicted in Fig. 4, where $G(s) = -\text{diag}\{k, k\}$ (a constant or instantaneous block). The disturbance signal d represents the walker's motion. In this case, V_w and Ω_w are in general both different from zero and the system kinematics becomes that described by eq. (2).

A persistent locomotion will in general prevent the convergence of the walker position to the platform center when using the control law (3–4). In fact, the closed-loop input-output equations become in this case

$$\dot{x} = -kx + V_{w,x}, \quad \dot{y} = -ky + V_{w,y}.$$

If the user walks indefinitely along a straight line with constant velocity \bar{V} , he/she will reach a steady-state position at a distance $\bar{R} = \bar{V}/k$ from the origin. In this case, from standard linear control analysis, the steady-state error can be completely eliminated by adding an integral action in the control loop before the disturbance entry point, i.e., by replacing eq. (4) with

$$v_1 = -k \left(x + a \int x dt \right), \quad v_2 = -k \left(y + a \int y dt \right), \quad (7)$$

for suitable $k > 0$ and $a > 0$. These proportional-integral (PI) gains can be chosen so as to assign desired closed-loop poles (two for each input-output channel) in the left-hand side of the complex plane. With reference to the equivalent linear scheme in Fig. 4, this PI action is realized by setting each diagonal component of $G(s)$ equal to $-k(s + a)/s$.

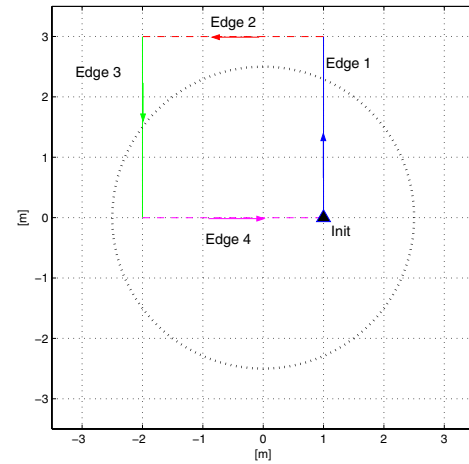


Fig. 5. Walker virtual locomotion: A square path executed counterclockwise starting from the Init point (the dotted circle represents the platform boundary)

The obtained *astatic* behavior copes with an unconstrained infinite walking of the user along a straight line. However, an overshooting of the controlled output is typically associated to the presence of the integral control term. For more general motion patterns, the control law of eqs. (3) and (7) may not lead to full recovery of the walker position to the origin. Nonetheless, in the absence of an estimate of the walker velocity, not much more can be done. The trade-offs of this modified controller will be illustrated in the comparative simulations reported in the next section.

V. SIMULATION RESULTS

The motion control law (3–4) and its modified version, with (7) in place of (4), have been tested on several motion patterns for the walker, including straight line, circular paths, and random walks executed at different speeds.

As a case study, we report here the results for the square path with 3 m sides shown in Fig. 5. The walker starts at rest from the ‘Init’ absolute position $(1, 0)$ and moves along each edge with a trapezoidal velocity profile, having symmetric acceleration/deceleration phases at 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 $\pi/2 \text{ rad/s}$. Therefore, the total trajectory lasts 16 s . Note that, without motion control of the platform, the walker would exit from the boundary of the circular platform of radius 2.5 m .

The absolute motion of the walker under the platform control law (3–4), with $k_1 = k_2 = 2.5$, and the corresponding commands (v, ω) are shown in Fig. 6 and Fig. 7, respectively. The absolute orientation of the walker is displayed by a segment, while different patterns/colors are used for the virtual motion along each side of the square. As expected, each time the walker stops at a corner to perform a turn, he/she is pulled towards the origin along a straight line. The control commands are smooth and never exceed the walker's voluntary speed. Also, it can be recognized that a periodic behavior is reached starting with the third side of the square path.

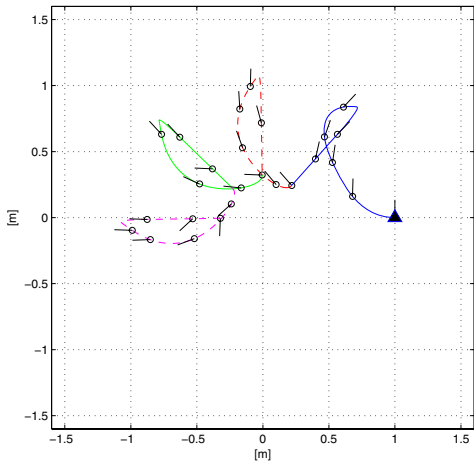


Fig. 6. Walker absolute locomotion under the platform controller (3–4): at start, axis X_t coincides with X_0

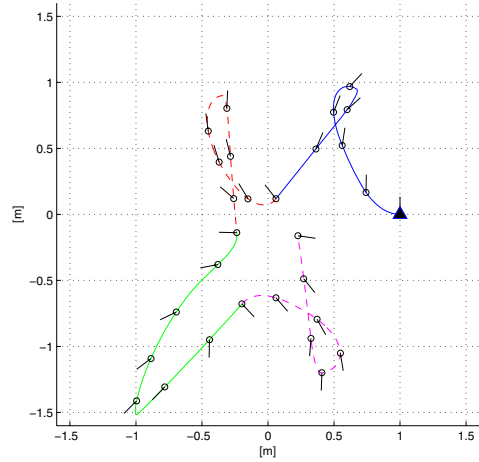


Fig. 8. Walker absolute locomotion under the platform controller (3) and (7): at start, axis X_t coincides with X_0

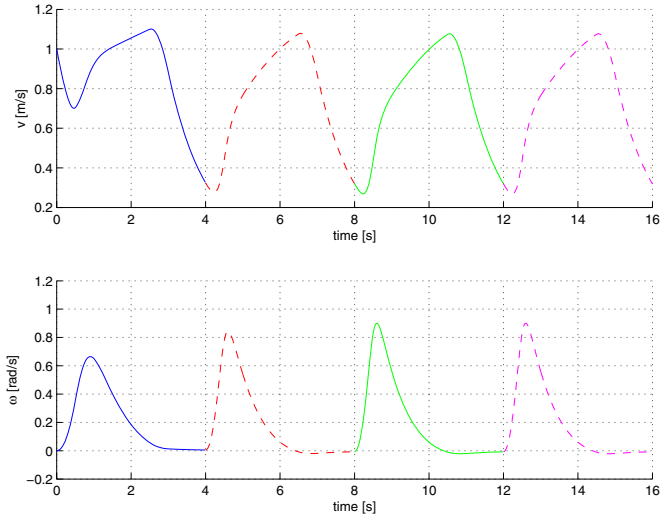


Fig. 7. Linear and angular velocity commands with controller (3–4)

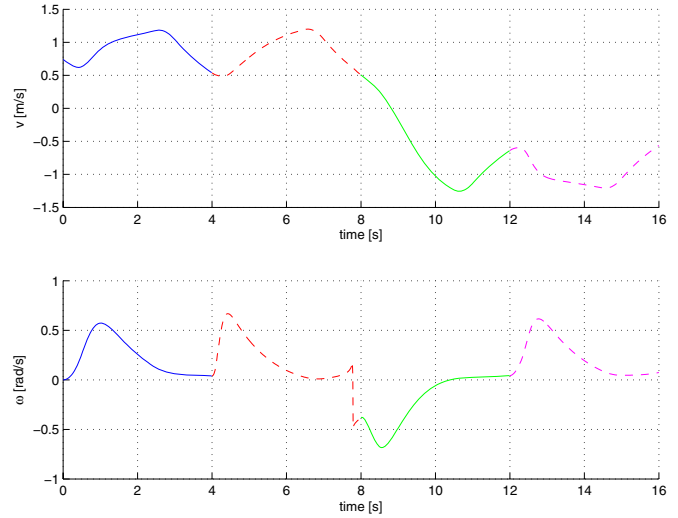


Fig. 9. Linear and angular velocity commands with controller (3) and (7)

The absolute motion of the walker under the platform controller of eqs. (3) and (7), with $k = 0.74$ and $a = 0.185$, and the corresponding commands (v, ω) are shown in Fig. 8 and Fig. 9, respectively. In this case, the linear velocity command is slightly larger whereas the peaks of the angular command are halved. The integral of $\omega(t)$ achieves an approximately zero average value along the whole path. Although the final position error is smaller than in the previous case, the position overshoot during intermediate motions is quite evident, with the walker coming closer to the platform boundary.

The comparison of the values of $\cos \alpha$ shown in Fig. 10 for the two controllers indicates two interesting facts. For the first control law, the platform is able to counteract the walker's motion while keeping him/her always in the same half-plane of the rotating frame (on the positive side of X_t , as in the starting position); this implies that control singularities are never met. On the other hand, the second control law undergoes a rapid angular velocity reversal just before $t = 8$ s, i.e., at the second

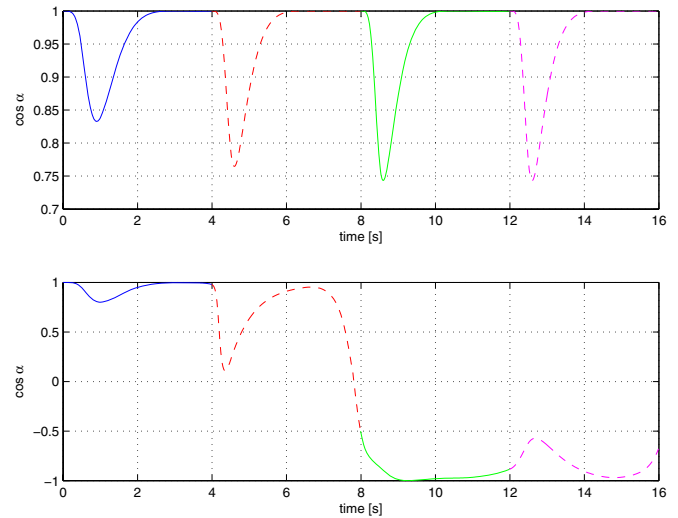


Fig. 10. Evolution of $\cos \alpha$ with controller (3–4) (top) and controller (3) and (7) (bottom)

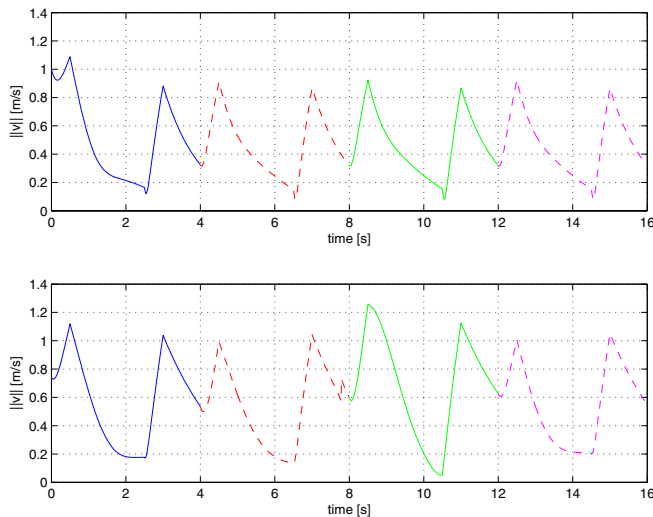


Fig. 11. Norm of the linear velocity felt by the walker during motion with controller (3-4) (top) and controller (3) and (7) (bottom)

corner of the path, with the walker crossing the Y_t -axis. From there on, the walker is pulled with a negative linear velocity v (see also Fig. 9) and the final equilibrium is reached with $\alpha = \pi$. Note also that there is some delay for v to become negative, due to the overshoot with the modified (integral) control law.

In order to assess the physiological effects on the user, Fig. 11 shows the norm of the linear velocity vector felt by the walker during the whole combined motion on the *CyberCarpet*. The two controllers behave in a similar way. From the last equation in (2), the angular velocity $\dot{\theta}_w$ of the walker can be determined from the plots of ω in Figs. 7 and 9 by adding the constant angular velocity Ω_w of the walker (which is different from zero only in correspondence to the turns at the corners).

A 3D graphical environment has been developed, using Simulink and Visual Nastran. Videos can be found at the website <http://www.dis.uniroma1.it/labrob/research/CW.html>.

VI. CONCLUSIONS

We developed a kinematic model for the *CyberCarpet* nonholonomic platform and designed a nonlinear feedback law that is able to safely keep the user close to the platform center, using only the available information on the position of the walker and on the angular position of the turntable. The control design is based on input-output decoupling and linearization of the cartesian motion of the walker, which allows the easy tuning of few control parameters on the linear side of the problem (a P or a PI error feedback loop).

One outcome of this initial study is that a relatively simple control strategy is effective in overcoming the restricted instantaneous mobility of the platform due to its nonholonomic nature. Although there are location of the walker on the turntable where his/her voluntary velocity cannot be instantaneously ‘canceled’ by any platform motion, the proposed

control law compensates over time for the walker locomotion using maneuvers driven by a pure sensory feedback.

The robust handling of potential control singularities and the consideration of explicit bounds on the actuated inputs are subject of current research, including the introduction of dead zones and command saturations. In the absence of measurements of the walker voluntary velocity, the addition of a feedforward term estimating on-line the walker’s velocity may prove beneficial in terms of control performance. This could be achieved, e.g., by using suitable observers for unknown system disturbances [18]. Other on-going control work takes as a starting point a second-order kinematic model, with linear and angular acceleration inputs, or a control-oriented dynamic model of the combined walker/platform system.

ACKNOWLEDGMENTS

Work supported by EU STREP FP6-511092 *CyberWalk*.

REFERENCES

- [1] H. Iwata, “Locomotion interface for virtual environments,” *9th Int. Symp. on Robotics Research*, pp. 275–282, 2000.
- [2] J. M. Hollerbach, “Locomotion interfaces,” *Handbook of Virtual Environments Technology*, (K.M. Stanney, Ed.), pp. 239–254, 2002.
- [3] EU Project FP6-511092 (*CyberWalk*), <http://www.cyberwalk-project.org>, 2005.
- [4] Y. Dupuis, C. Anthierens, J.-L. Impagliazzo, and L. Yushchenko, “Design of a sensorial and driving locomotion interface,” *16th IFAC World Congr.*, Praha, CZ, July 4–8, 2005.
- [5] R. C. Hayward and J.M. Hollerbach, “Implementing virtual stairs on treadmills using torso force feedback,” *2002 IEEE Int. Conf. on Robotics and Automation*, pp. 586–591, 2002.
- [6] H. Iwata, H. Yano, and F. Nakaizumi, “Gait Master: A versatile locomotion interface for uneven virtual terrain,” *IEEE Virtual Reality Conf.*, pp. 131–137, 2001.
- [7] R. Darken, W. Cockayne, and D. Carmein, “The Omnidirectional Treadmill: A locomotion device for virtual worlds,” *Symp. User Interface Software and Technology*, pp. 213–221, 1997.
- [8] H. Iwata, “The Torus Treadmill: Realizing locomotion in VEs,” *IEEE Computer Graphics and Applications*, vol. 9, no. 6, pp. 30–35, 1999.
- [9] H. Iwata, H. Yano, H. Fukushima, and H. Noma, “CirculaFloor,” *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 64–67, 2005.
- [10] Z. Wang, K. Bauernfeind, and T. Sugar, “Omni-directional treadmill system,” *11th IEEE Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 367–373, 2003.
- [11] A. Nagamori, K. Wakabayashi, and M. Ito, “The Ball Array Treadmill: A locomotion interface for virtual worlds,” *Work. on New Directions in 3D User Interfaces (at VR 2005)*, Bonn, D, May 12–16, 2005.
- [12] 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, no. 1, pp. 27–35, 1995.
- [13] C. Samson, “Time-varying feedback stabilization of car-like wheeled mobile robots,” *Int. J. of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.
- [14] C. Samson, “Control of chained systems. Application to path following and time-varying point-stabilization of mobile robots,” *IEEE Trans. on Automatic Control*, vol. 40, no. 1, pp. 64–77, 1995.
- [15] M. K. Bennani and P. Rouchon, “Robust stabilization of flat and chained systems,” *3rd European Control Conf.*, pp. 2642–2646, 1995.
- [16] O. J. Sørđalen and O. Egeland, “Exponential stabilization of nonholonomic chained systems,” *IEEE Trans. on Automatic Control*, vol. 40, no. 1, pp. 35–49, 1995.
- [17] A. De Luca and G. Oriolo, “Local incremental planning for non-holonomic mobile robots,” *1994 IEEE Int. Conf. on Robotics and Automation*, pp. 104–110, 1994.
- [18] A. De Luca, R. Mattone, and P. Robuffo Giordano, “Feedback/feedforward schemes for motion control of the *CyberCarpet*,” *2006 IFAC Symp. on Robot Control*, Bologna, I, 2006.