FP6-511092

CyberWalk

The CyberCarpet - Enabling Omni-directional Walking in Virtual Worlds

Instrument: STREP
Thematic Priority: IST

Deliverable T5.1/D1

Mobility Analysis and Simulation

Due date of deliverable: 31.03.2006
Actual submission date: 15.04.2006 updated version 04.12.2006

Start date of project: 01.04.2005 Duration: 36 month

Organisation name of lead contractor for this deliverable: **UOR**
Revision: **2**

PROJECT CO-FUNDED BY THE EUROPEAN COMMISSION WITHIN THE 6. FRAMEWORK PROGRAMME (2002–2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contents Page

Summary

This document is a revised version of deliverable T5.1/D1, taking into account the review comments from the Project Officer and Reviewers, resulting from the 1st Review Meeting, held in Rome on June 7, 2006. In particular, the mobility analysis and simulation activity already performed for the *CyberCarpet* "ball-array" concept (see Fig. 2) has been now extended (whenever possible) and/or suitably completed, in order to cover also the new "omni-directional" design concept (see Fig. 4) recently considered within the project. In order to keep the report structure simple and clear, each section of the report has been decomposed into two subsections where the two carpet designs are separately treated.

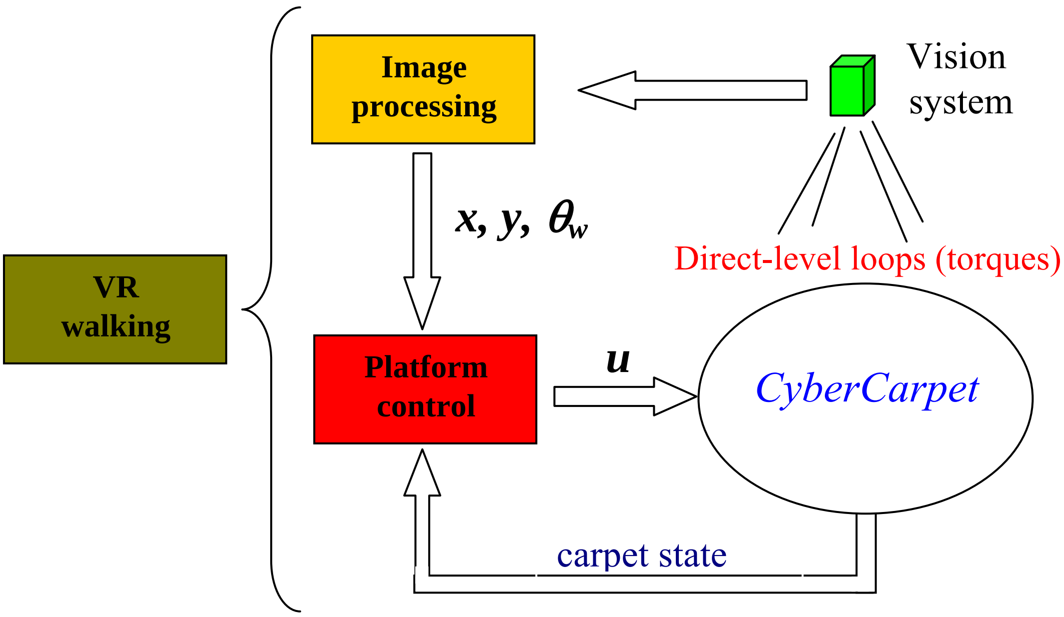

Fig. 1: The general *CyberCarpet* system architecture: When the ball-array carpet is of concern, the control input vector u is constituted by the linear and angular carpet velocities v and ω, and the angular carpet orientation θ is fed back to the controller; when instead the omni-directional carpet is considered, the components of u are the linear velocities v_x and v_y, and no feedback from the carpet position to the controller is necessary

For the treadmill/walker system, and for both treadmill design concepts, first- and second-order kinematic modelling are illustrated in Sects. 1 and 2, respectively, while the complete dynamic model (i.e., considering as inputs the generalized forces actuating the platform) is given in Sect. 3. These models are mainly intended for motion control design, with the system architecture shown in Fig. 1. The evaluation of the kinematic/dynamic effects of the platform motion on the walker is performed in Sect. 4. Finally, the overall simulation setup, developed using Simulink, is briefly described in Sect. 5, where results for representative walker motions are also given.

1. First-order kinematic modelling of the platform/walker system

1.1. Ball-array concept

The *CyberCarpet* “ball-array” concept is sketched in Fig. 2: a belt and a turntable transmit linear and angular motion to a walker on top of the carpet through a passive ball-array board. The rotating balls are fitted into the array board and are in contact with the moving belt on the bottom side, so that an object on the top side of the board moves in the opposite direction of the corresponding point on the belt.

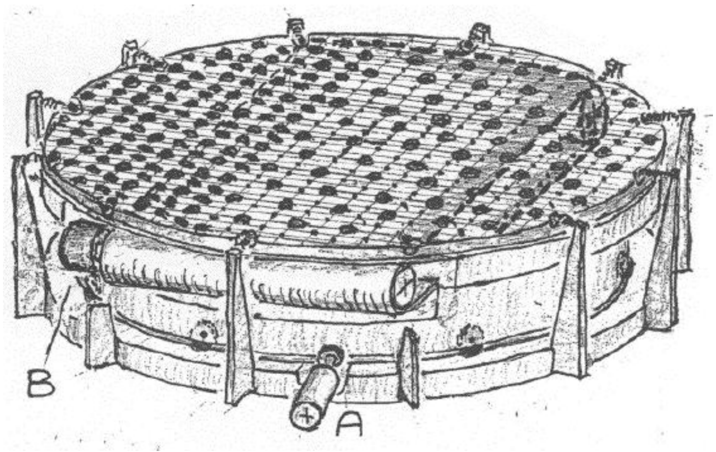

Fig. 2: The ball-array concept for the CyberCarpet

A first-order kinematic model of the *CyberCarpet* (i.e., assuming as system inputs the linear and angular velocities v and ω of the treadmill) can be derived with the help of Fig. 3.

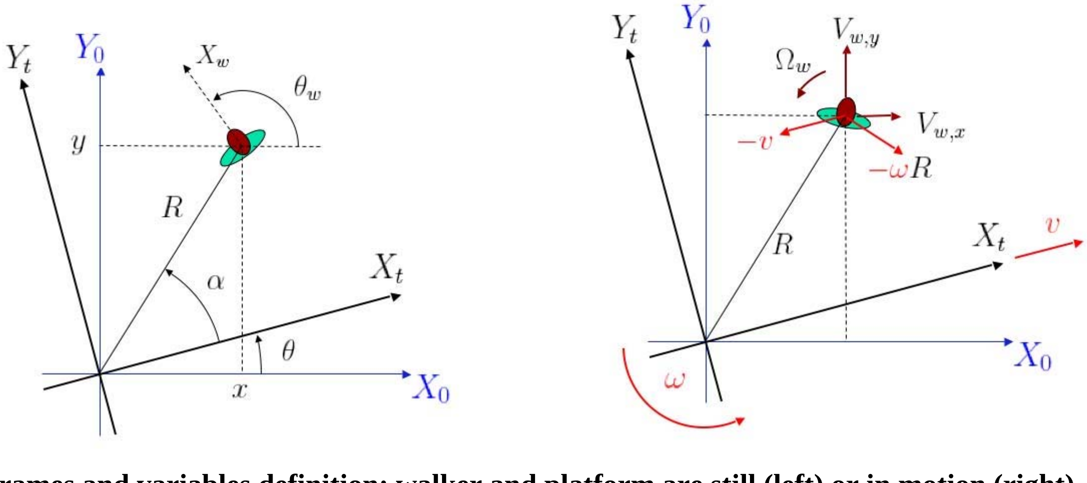

Fig. 3: Frames and variables definition: walker and platform are still (left) or in motion (right)

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 (on the bottom side of the ball-array board), with the X_t–axis in the direction of the belt (along which linear motion is actuated). Both frames have the origin at the centre of the *CyberCarpet*. In the left side of Fig. 3 the X_w axis (directed as the walker’s sight) of the frame attached to the walker is also displayed (the Z_w axis is directed as Z_0 and Y_w is consequently defined): this frame will be used in Sect. 4 to describe the kinematic and dynamic effects of the platform motion on the user. The absolute position and orientation of the walker (standing on the top side of the ball-array board) are, respectively, (x, y) and θ_w, while R is his/her distance from the centre. The angle α = atan2(x, y) - θ locates the position of the walker in the rotating frame (X_t, Y_t). Due to the presence of the ball-array surface, any actuated motion of the belt, on the bottom of the ball array, will result in a reverse motion imposed to the walker standing on top of the ball array, i.e., a forward motion command v (see Fig. 3, right) will move the user backwards (i.e., in the direction $-v$), and a clockwise rotation of the turntable will turn the user counter-clockwise (i.e., the variations of θ and θ_w will have opposite signs). As a result, when the walker is standing still, we obtain

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

being v and ω the linear and angular velocity of the *CyberCarpet*, i.e., the values of the control inputs on the bottom of the ball-array board.

Note that the two Cartesian coordinates (x,y) may be replaced by suitable polar coordinates, e.g., the walker's distance R from the centre ($R^2 = x^2 + y^2$) and the angle α. One obtains

$$\begin{aligned}\dot{R} &= \frac{x\dot{x} + y\dot{y}}{R} = -v\frac{x\cos\theta + y\sin\theta}{R} = -v\frac{R\cos\alpha}{R} = -v\cos\alpha, \\ \dot{\alpha} &= \frac{x\dot{y} - y\dot{x}}{R^2} - \omega = v\frac{R\sin\alpha}{R^2} - 2\omega = v\frac{\sin\alpha}{R} - 2\omega.\end{aligned} \tag{2}$$

Simple analysis of kinematic eqs. (1) shows that a holonomic constraint exists, i.e., $\theta + \theta_w = const$, so that only one of these two variables can be independently controlled. This will not be 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 differential constraint

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

The nonholonomic nature of this constraint may be proved by standard nonlinear controllability analysis. As a result, the walker can be in principle moved to the origin $(x, y, \theta) = (0,0,0)$, by "suitable manoeuvres", for any initial system configuration. From eqs. (2), the following simple command sequence is sufficient: *i)* rotate the platform (ω = constant and $v = 0$) until the angle α becomes zero; *ii)* set $\omega = 0$ and use a (positive) control input v to drive R to zero; *iii)* rotate the carpet again (ω = constant and $v = 0$), until θ becomes zero. The third step may be used to bring θ_w instead of θ to zero, but not both to zero unless in the initial condition it was $\theta(0)+\theta_w(0) = 0$.

On the other hand, there exist singular configurations where only a restricted set of velocity directions are instantaneously allowed. This is clear, again, from the analysis of eqs. (2): when $\alpha = \pm\pi/2$, a displacement of the walker in the radial direction cannot be realized by any instantaneous velocity input command.

In view of these considerations, the motion control problem for the *CyberCarpet* is similar to that of 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, however, when the unconstrained walker starts to move.

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

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

where $(V_{w,x}, V_{w,y})$ and Ω_w are, respectively, the *intended* linear and angular walker velocities, expressed in the *absolute* frame (X_0, Y_0, Z_0) (see Fig. 3, right). Note that the velocity $V_w = [V_{w,x}\ V_{w,y}]^T$ in eq. (3) is related to ${}^wV_w$, i.e., the same velocity expressed in the frame (X_w, Y_w, Z_w) *attached to the walker*, by the relationship

$$V_w = \begin{bmatrix} \cos\theta_w & -\sin\theta_w \\ \sin\theta_w & \cos\theta_w \end{bmatrix} {}^wV_w = Rot(\theta_w)\, {}^wV_w\,. \tag{4}$$

Similarly, if ${}^tV_w$ is the intended walker velocity, expressed in the *virtual world* frame (X_t, Y_t, Z_t) attached to the treadmill, it holds

$$V_w = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} {}^tV_w = Rot(\theta)\, {}^tV_w\,. \tag{5}$$

The walker's velocities are assumed to be not directly measurable and will act as disturbances in the control system. Instead, the whole system state (x, y, θ, θ_w) can be assumed to be available, being (x, y, θ_w) provided by a high-level visual tracker and θ by an encoder on the turntable rotating axis (see Fig. 1).

1.2. Omni-directional concept

The *CyberCarpet* "omni-directional" concept is sketched in Fig. 4: an array of synchronous linear carpets C_1, C_2, ..., C_N can be shifted with velocity v_x in the (blue) direction that is orthogonal to the common (yellow) velocity v_y of each carpet, so that a velocity with any (green) direction in the plane can be obtained as a combination of the two motions.

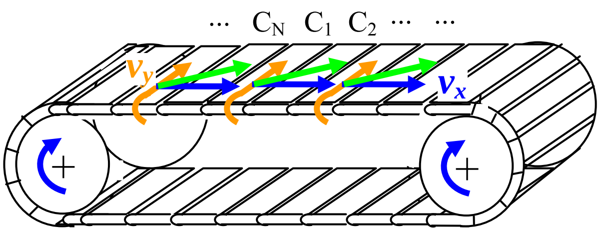

Fig. 4: The omni-directional concept for the CyberCarpet

Using the same notation as in Sect. 1.1, the first-order kinematic model of the *CyberCarpet* is in this case

$$\begin{aligned} \dot{x} &= v_x + V_{w,x} \\ \dot{y} &= v_y + V_{w,y} \\ \dot{\theta}_w &= \Omega_w \end{aligned} \tag{6}$$

where the input commands are now the linear treadmill velocities v_x and v_y (no motion inversion is present). Note that we have in this case one less state variable than in the ball-array model of eq. (3), since the treadmill and absolute frames are always parallel and we do not need the angle θ to characterize the treadmill state. Furthermore, the kinematics of eq. (6) does not exhibit any nonholonomic behaviour and the variable θ_w is not controllable, since it is not affected by any of the two available input commands[1] v_x and v_y: in par-

[1] Indeed, this is only true in the case of symmetrical contact between walker and carpet (both feet on the ground): in fact, when just one foot is on the ground, the contact force between foot and carpet results, in general, in a nonzero torque around the walker centre of mass (see Sect. 4).

ticular, when the walker is standing still, it holds θ_w = constant. However, this is not relevant, since we are only interested in controlling the walker position (x ,y).

2. Second-order kinematic modelling of the platform/walker system

2.1 Ball-array concept

If we assume that the ball-array treadmill motion is commanded through the linear and angular acceleration inputs a and η, the first-order kinematic model (3) is extended to

$$\begin{aligned}
\dot{x} &= -v\cos\theta + y\omega + V_{w,x} \\
\dot{y} &= -v\sin\theta - x\omega + V_{w,y} \\
\dot{\theta} &= \omega \\
\dot{\theta}_w &= -\omega + \Omega_w \\
\dot{v} &= a \\
\dot{\omega} &= \eta
\end{aligned} \tag{7}$$

which corresponds to adding two integrators on the velocity inputs of model (3) (see also Sect. 5). These two velocities become now two further states of the system. Therefore, in order to assume that the whole system state is available for feedback control design, also the linear and angular treadmill velocities v and ω must be measurable. Finally, note that this second-order kinematic model is characterized by the same motion singularities of the first-order model.

2.2 Omni-directional concept

As for the ball-array concept, the second-order kinematic model of the omni-directional treadmill is obtained by extending the first-order model (6) to

$$\begin{aligned}
\dot{x} &= v_x + V_{w,x} \\
\dot{y} &= v_y + V_{w,y} \\
\dot{\theta}_w &= \Omega_w \\
\dot{v}_x &= a_x \\
\dot{v}_y &= a_y
\end{aligned} \tag{8}$$

where the linear carpet accelerations a_x and a_y are the new control inputs, while the velocities v_x and v_y are now two further state variables.

3. Dynamic modelling of the platform/walker system

In the real *CyberCarpet* system, the carpet accelerations will be the result of generalized forces provided by suitable actuators, moving the ensemble of treadmill and walker. The walker is considered here as a passive payload[2] located at (x, y) on the treadmill (i.e., $V_{w,x} = V_{w,y} = 0$ in eqs. (7-8)), and having mass m_w and inertia I_w around its vertical axis. Friction is not modelled, but it is assumed to be sufficiently high to avoid any slipping at contact points.

3.1 Ball-array concept

The dynamic model of the platform/walker system has been computed in this case following the Lagrangian approach. Since no conservative forces/torques affect the system, the Lagrangian function equals the kinetic energy T of the system, which is the sum of the two terms T_c and T_w related to the carpet and the walker, respectively. Before computing these two terms, note that only part of the treadmill will translate under the ef-

[2] Modelling the dynamics of human walking is out of the scope of this report, and the walker's motion is still treated here as a disturbance acting at the velocity level.

fect of the linear force F, namely the *carpet* (having mass m_c always centred at the platform origin), while the whole treadmill (*carpet* + *turntable*, having total inertia I_c around the actuated rotation axis) will rotate under the effect of the torque τ. Furthermore, the absolute linear and angular velocities and acceleration of the walker have the same amplitude and direction, but opposite sign, than those of the corresponding (bottom) point on the belt. In other words, the ball-array board is treated as an ideal transmission device.
Then, it is

$$
\begin{aligned}
T_c &= \frac{1}{2}m_c v^2 + \frac{1}{2}I_c\omega^2 = \frac{1}{2}m_c\left((\dot{x} - y\omega)^2 + (\dot{y} + x\omega)^2\right) + \frac{1}{2}I_c\omega^2 \\
T_w &= \frac{1}{2}m_w\left(\dot{x}^2 + \dot{y}^2\right) + \frac{1}{2}I_w\omega^2
\end{aligned}
\tag{9}
$$

and dynamic model can be written as

$$
\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right)^T - \left(\frac{\partial T}{\partial q}\right)^T = \Phi \,, \tag{10}
$$

where $q = [x \quad y \quad \theta]$, $\dot{q} = [\dot{x} \quad \dot{y} \quad \omega]^T$, $T(q,\dot{q}) = T_c(q,\dot{q}) + T_w(\dot{q}) = \dot{q}^T B(q)\dot{q}$ is the total kinetic energy, with $B(q)$ positive definite, and Φ is the vector of generalized forces that perform work on the components of q.
In order to find the expression of Φ as a function of $u = [F \ \tau]^T$ and q, we use eq. (1), which can be rewritten as

$$
\dot{q} = G(q)\begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{11}
$$

By virtue of the kineto-static relationship between generalized velocities and forces, this implies

$$
u = G(q)^T\Phi,
$$

and thus

$$
\Phi = \left(G(q)^T\right)^{\#} u + A(q)\cdot\lambda \,, \tag{12}
$$

where $\left(G(q)^T\right)^{\#} = G(q)\left(G(q)^T G(q)\right)^{-1}$ is the right pseudoinverse of matrix $G(q)^T$, $A(q)$ is a 3×1 basis of the null space of $G(q)^T$ (hence, $G(q)^T A(q) = 0$), and λ is a real parameter.

Using eqs. (9), (11) and (12), and the time derivative of (11), into eq. (10), and left-multiplying both members by $G(q)^T$, one gets (dropping dependencies for simplicity)

$$
G^T BG\begin{bmatrix} a \\ \eta \end{bmatrix} + \left(G^T B\dot{G} + G^T \dot{B}G\right)\begin{bmatrix} v \\ \omega \end{bmatrix} - G^T\frac{\partial T}{\partial q} = \begin{bmatrix} F \\ \tau \end{bmatrix},
$$

i.e.,

$$
\begin{aligned}
F &= (m_c + m_w)a - m_w R\sin\alpha\cdot\eta + m_w R\omega^2\cos\alpha, \\
\tau &= \left(I_c + I_w + m_w R^2\right)\eta - m_w R\sin\alpha\cdot a,
\end{aligned}
\tag{13}
$$

which extends the second-order kinematic model (7) and can be more compactly rewritten as

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = B(R,\alpha)\begin{bmatrix} a \\ \eta \end{bmatrix} + n(R,\alpha,\omega),$$

with matrix $B(R,\alpha)$ positive definite.

In principle (i.e., assuming in particular that accurate estimates of the walker mass and inertias are available), eqs. (13) can always be decoupled and linearized by choosing the force/torque inputs as

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = B(R,\alpha)\begin{bmatrix} u_a \\ u_\eta \end{bmatrix} + n(R,\alpha,\omega), \tag{14}$$

being u_a and u_η two auxiliary inputs. This choice results in

$$\dot{v} = a = u_a, \qquad \dot{\omega} = \eta = u_\eta .$$

Thus, from the point of view of control design, models (7) and (13) are equivalent, and eqs. (13) will be used in the following only to evaluate the treadmill actuators effort in response to a motion (*inverse dynamics*).

3.2 Omni-directional concept

The dynamic model of the system is in this case much simpler, since no platform rotations are present, and the walker is directly coupled with the carpet (i.e., his/her motion is not reversed through the presence of the ball-array). As a consequence, one gets

$$\begin{aligned} F_x &= (m_c + m_w)\, a_x \\ F_y &= (m_c + m_w)\, a_y \end{aligned} \tag{15}$$

where F_x and F_y are the actuator forces applied along the x and y directions, respectively. As expected, the dynamic model is linear and fully decoupled.

4. Kinematic/dynamic effects of platform commands 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 walker moves with constant velocity in the virtual world, she/he will feel "apparent" accelerations 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. In this section, we treat the ball-array and omni-directional designs in a unified framework, since the more general formulas of the ball-array treadmill also apply to the omni-directional case when it is set $\omega = \eta = 0$, and $\dot{v} = a = \sqrt{a_x^2 + a_y^2}$, $\theta = \mathrm{atan2}(a_y, a_x)$.

When the user walks at constant velocity ${}^{w}V_w$ in the non-inertial virtual world[3], the total *apparent* acceleration that she/he feels equals her/his absolute acceleration (computable by analytic differentiation of the first two equations in (7) or (8)), *changed in sign*. This 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 *relative* velocity (*Coriolis* acceleration). All these components should be expressed in the frame (X_w, Y_w, Z_w) attached to the walker, in order to evaluate the physiological effects on the user. The results of the described computation procedure are reported below (three-dimensional vectors are denoted by an arrow on top).

- Inertial acceleration (due to the linear and angular accelerations of the reference frame):

[3] Note that, for the case of ball-array platform, a constant velocity ${}^{w}V_w$ in the virtual world does not correspond to constant absolute velocities ($V_{w,x}$, $V_{w,y}$) in eq. (7), see also eq. (4).

$$\vec{a}_{in} = \dot{v}\begin{bmatrix}\cos(\theta-\theta_w)\\ \sin(\theta-\theta_w)\\ 0\end{bmatrix} + \dot{\omega}R\begin{bmatrix}-\sin(\theta+\alpha-\theta_w)\\ \cos(\theta+\alpha-\theta_w)\\ 0\end{bmatrix} = a\begin{bmatrix}\cos(\theta-\theta_w)\\ \sin(\theta-\theta_w)\\ 0\end{bmatrix} + \eta R\begin{bmatrix}-\sin(\theta+\alpha-\theta_w)\\ \cos(\theta+\alpha-\theta_w)\\ 0\end{bmatrix}$$

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

- centrifugal component (due to the frame rotation)

$$\vec{a}_{cen} = \omega^2 R\begin{bmatrix}\cos(\theta+\alpha-\theta_w)\\ \sin(\theta+\alpha-\theta_w)\\ 0\end{bmatrix} = \omega^2 Rot(-\theta_w)\begin{bmatrix}x\\ y\end{bmatrix}, \quad \text{with } \|\vec{a}_{cen}\| = \omega^2 R;$$

- Coriolis component (due to the coupling between the walker relative velocity and the rotation of the non-inertial frame)

$$\vec{a}_{Cor} = 2\vec{\omega} \times {}^w\vec{V}_w,$$

  where ${}^w\vec{V}_w$ is the walker intended velocity, expressed in her/his frame.

Note that, for the inertial and centrifugal components we have also given a (computable) expression that does not depend on the angle α (not defined at the origin).
Beside each linear acceleration component $\vec{a}_k$ described above (k = *in*, *cen*, *Cor*), an angular acceleration $\vec{\eta}_k$ must be also considered, resulting from the momentum of the corresponding apparent force, applied at the walker's center of mass, around the feet contact point on the *CyberCarpet* floor. If the walker is schematized as a homogeneous cylinder of height h, it holds

$$\vec{\eta}_k = \frac{3}{2h^2}\vec{a}_k \times \vec{h},$$

where $\vec{h}$ is the vector from the walker's feet to his/her head. Thus, under the assumption that the walker is standing in a vertical position (i.e., $\vec{h}$ is directed along the Z axis), each component $\vec{\eta}_k$ lies on the horizontal plane (in particular, it is rotated by π/2 rad w.r.t. the corresponding linear acceleration[4] $\vec{a}_k$), and its amplitude is proportional to that of $\vec{a}_k$ by a factor that depends on the walker height.
Finally, in the case when the contact points of the walker on the floor are not symmetrical with respect to the walker centre of mass (e.g., only one foot is on the ground), we also have to consider the angular acceleration caused by the moment of the contact force on the floor around the vertical axis passing through the walker centre of mass. This amounts to

$$\vec{\eta}_{cont} = \frac{m_w}{I_w}[\ddot{x} \quad \ddot{y}]^T \times \vec{d},$$

where $\ddot{x}$ and $\ddot{y}$ are computed by differentiating the first two equations in (7) or (8), and $\vec{d}$ is the vector from the contact point at the foot to the walker centre of mass (see Fig. 5).

[4] In other words, the X_w component of $\vec{a}_k$ causes a (proportional) angular acceleration directed as Y_w, and the Y_w component of $\vec{a}_k$ causes an angular acceleration directed as $-Y_w$.

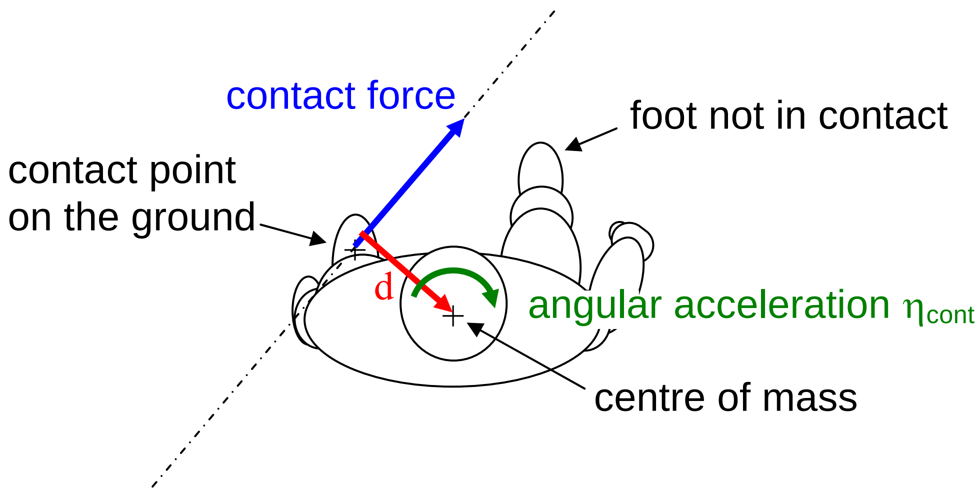

Fig. 5: The origin of the angular acceleration η_{cont} when the contact points on the ground are not symmetrical with respect to the centre of mass

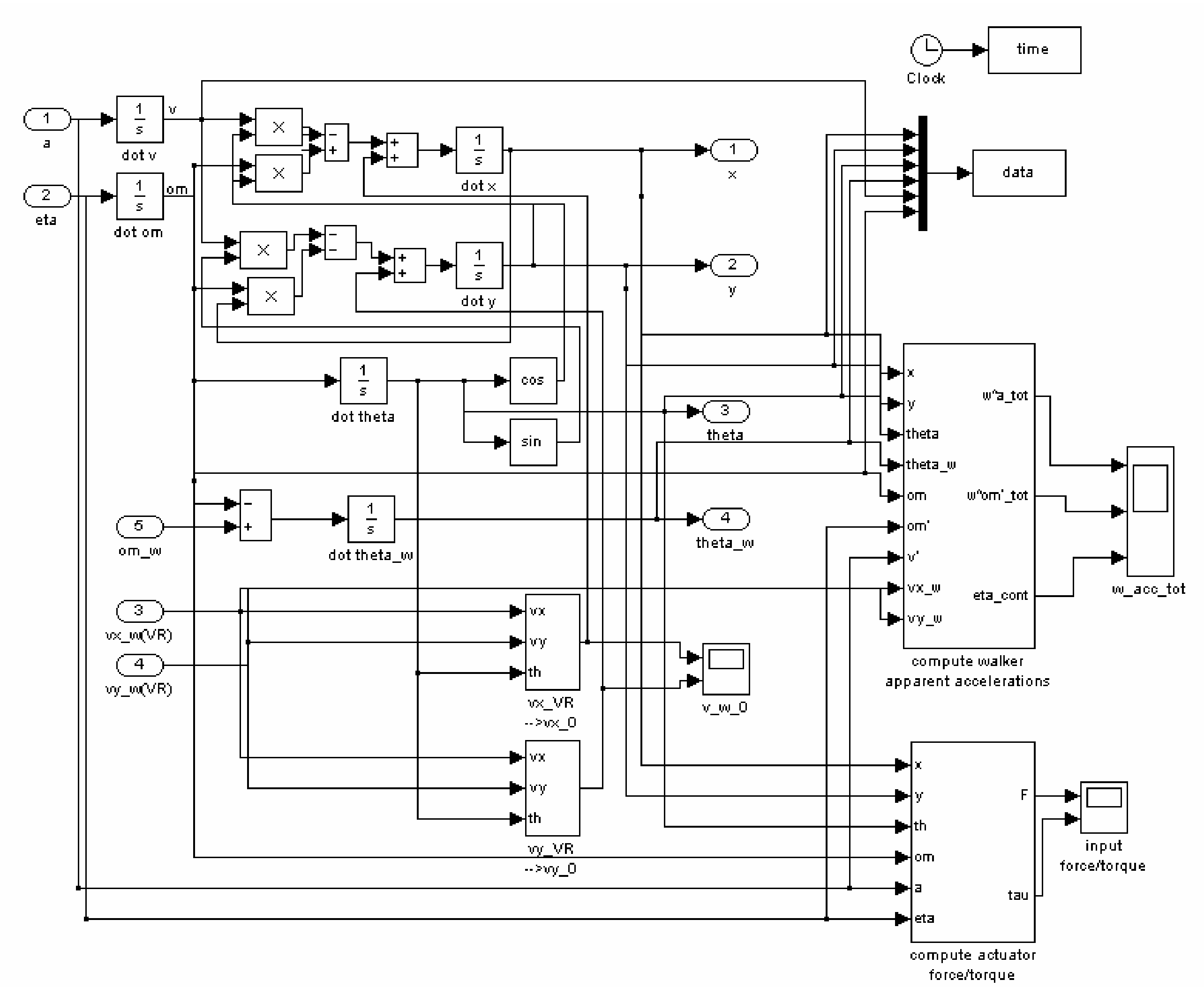

Fig. 6: Ball-array design: Simulink model for the overall carpet/walker system

5. Simulation set-up

5.1 Ball-array concept

The complete second-order model (7) of the ball-array carpet/walker system has been implemented and simulated in Simulink®. The corresponding files *BallArrayModel.mdl* and *InitBallArrayModel.m* have been provided to the project partners, while a short user manual for the software module is reported in Appendix A.

The model diagram is represented in Fig. 6. Note that two submodels are present in the scheme, labelled as *compute actuator force/torque* and *compute walker apparent accelerations*, performing the computations described in Sect. 3 and Sect. 4, respectively.

We report here some representative simulation results for the case when a bang-coast-bang angular acceleration command (of amplitude 2 rad/s^2, and duration T_{bang} = 0.7 s and T_{coast} = 0.6 s) is applied (in open-loop) to the treadmill, while the walker moves in the X_w direction with a bang-coast-bang linear acceleration profile (of amplitude 4 m/s^2, and duration T_{bang} = 0.5 s and T_{coast} = 1 s) starting from the point $(x,y) = (0, 1)$, with $\theta_w = \pi/2$, and being initially at rest. The corresponding kinematic/dynamic effects on the user, in terms of felt velocities and accelerations, all expressed in the walker frame, are reported in Fig. 7-Fig. 11.

In particular, Fig. 7 displays the absolute linear and angular velocities of the walker in the plane (thus, two linear and one angular component), including the contribution by the walker locomotion, while Fig. 8 reports just the contributions by the carpet motion. Furthermore, Fig. 9-Fig. 11 show the inertial, centrifugal and Coriolis components of the apparent *linear* accelerations felt by the walker. All these accelerations lay on the horizontal plane, thus only their X_w and Y_w components have been reported. As a consequence of the trivial relationship between these linear accelerations and the corresponding components of the apparent angular acceleration felt by the walker (see footnote 4), the latter have not been reported. Finally, Fig. 12 displays the angular acceleration η_{cont}, directed along the vertical axis, due to a one-foot contact (the human walk has been modelled as a sequence of steps of fixed length 0.7 m, with frequency proportional to the walker velocity).

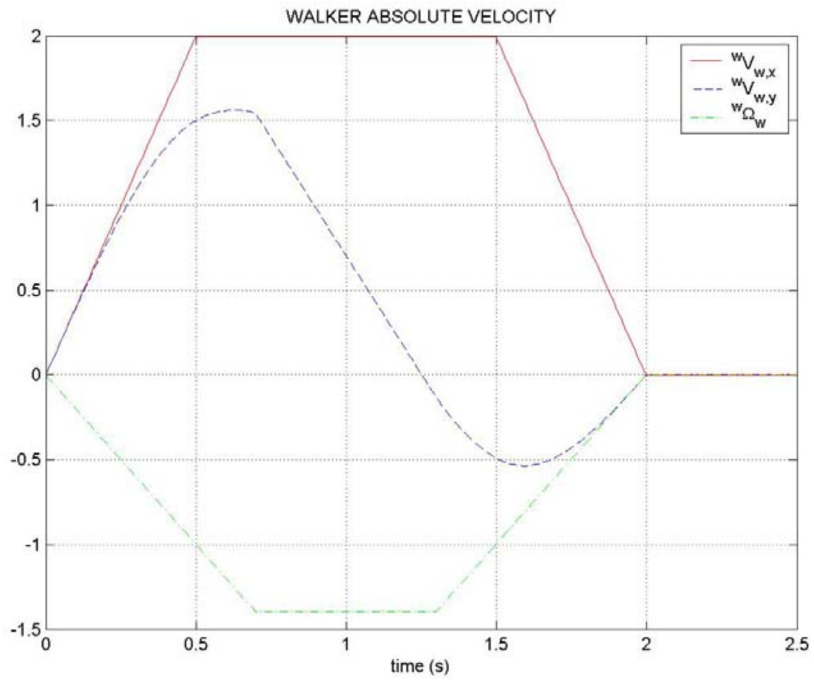

Fig. 7: Ball-array design: Absolute linear and angular walker velocity, expressed in the walker frame

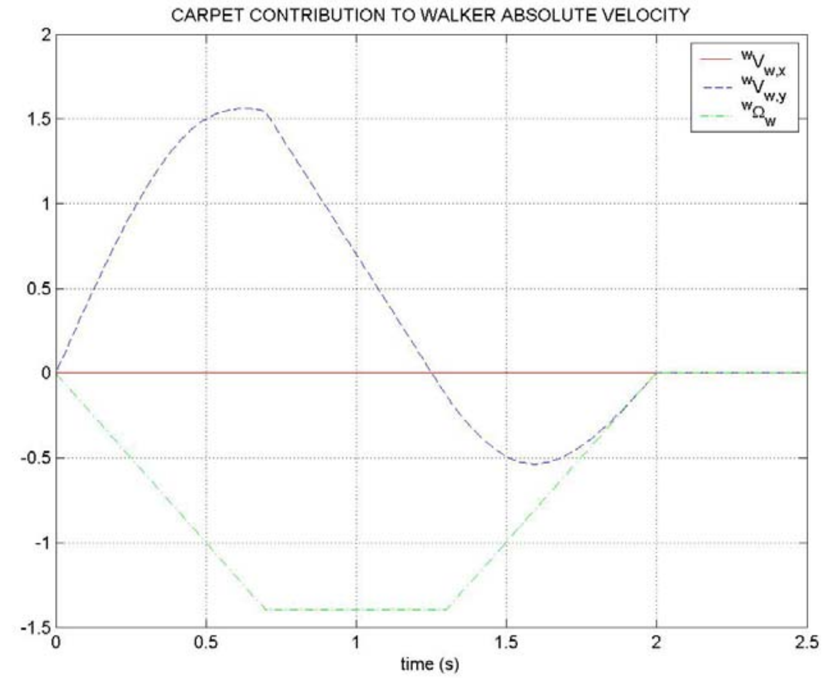

Fig. 8: Ball-array design: Contribution of carpet motion to walker absolute velocity

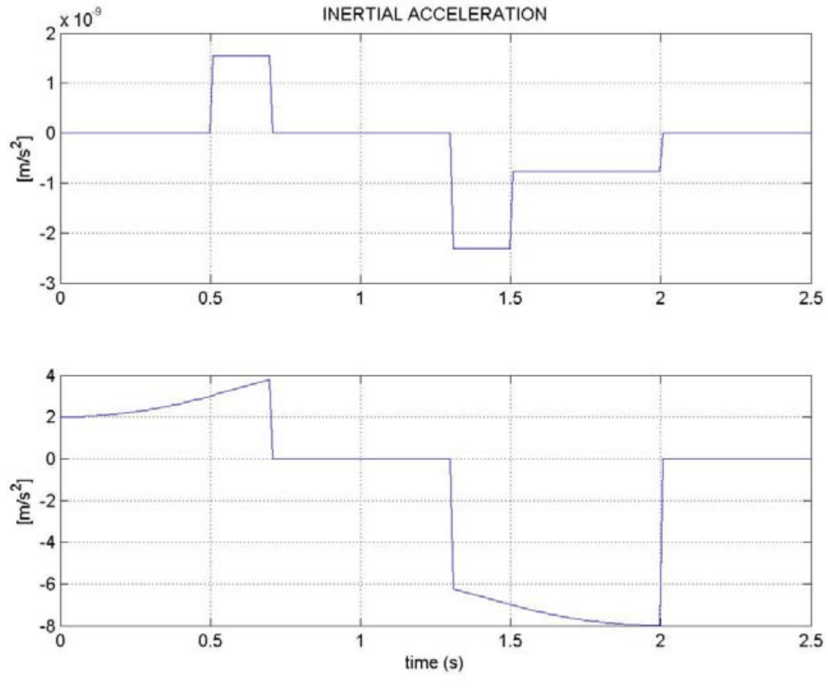

Fig. 9: Ball-array design: Inertial acceleration felt by the walker in the X_w (top) and Y_w (bottom) directions

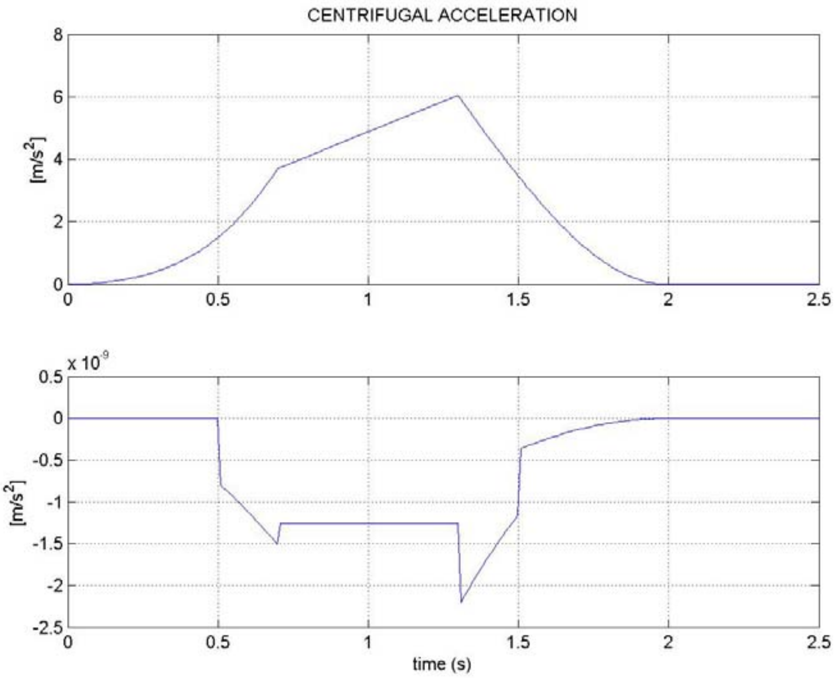

Fig. 10: Ball-array design: Centrifugal acceleration felt by the walker in the X_w (top) and Y_w (bottom) directions

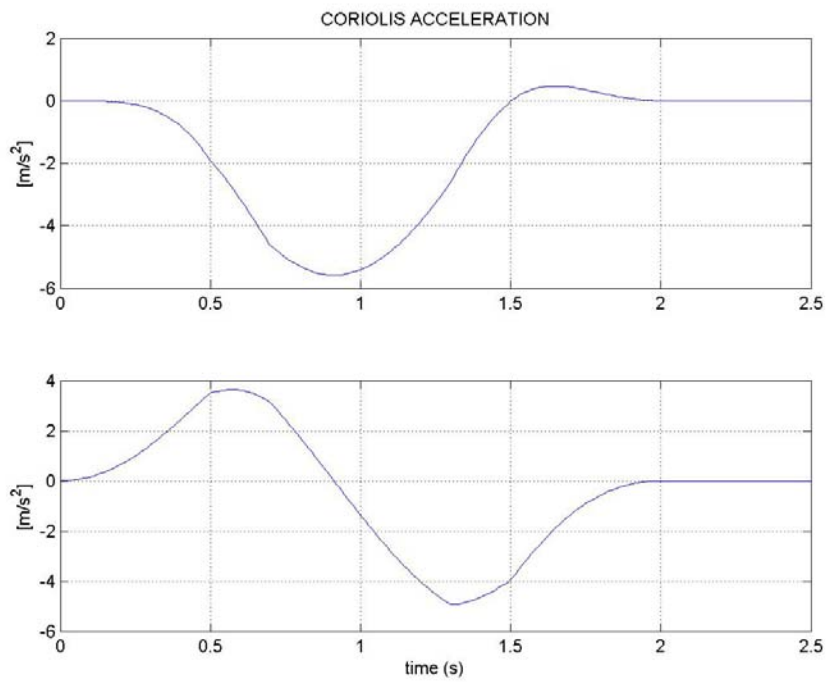

Fig. 11: Ball-array design: Coriolis acceleration felt by the walker in the X_w (top) and Y_w (bottom) directions

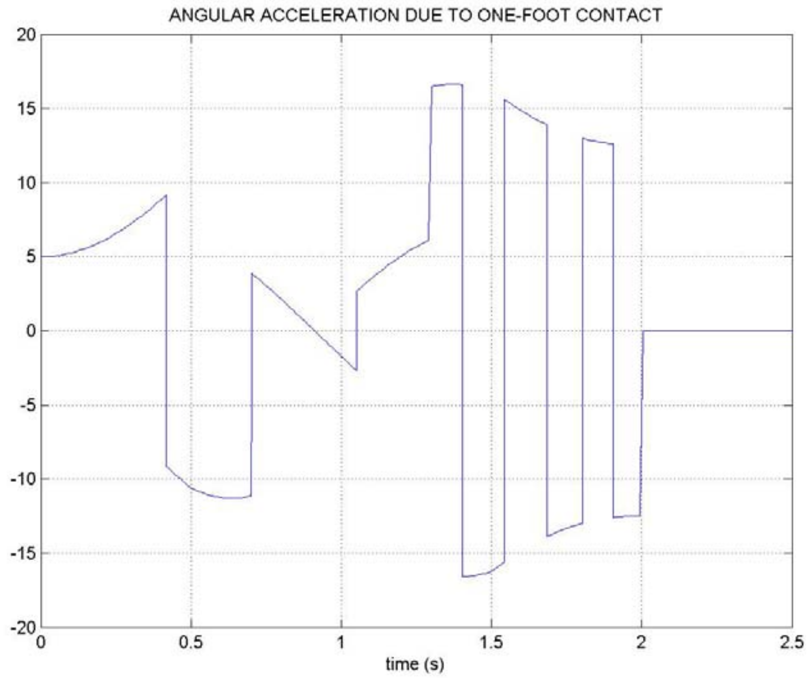

Fig. 12: Ball-array design: Angular acceleration η_{cont} felt by the walker along the vertical axis, due to one-foot contact

5.2 Omni-directional concept

The complete second-order model (8) of the omnidirectional carpet/walker system has been implemented and simulated in Simulink®. The corresponding files *OmniDirectionalModel.mdl* and *InitOmniDirectionalModel.m* have been provided to the project partners, while a short user manual for the software module is reported in Appendix B.

The model diagram is represented in Fig. 13. As for the ball-array model, a submodel *compute walker apparent accelerations*, performing the computations described in Sect. 4, is also present.

We report here some representative simulation results for the case when bang-coast-bang acceleration commands (of amplitude 2 rad/s^2, and duration T_{bang} = 0.7 s and T_{coast} = 0.6 s) are applied at the same time (in open-loop) along the X and Y directions, while the walker moves in the X direction with a bang-coast-bang linear acceleration profile (of amplitude 2 m/s^2, and duration T_{bang} = 0.5 s and T_{coast} = 1 s) starting from the point (x,y) = (0, 1), being initially at rest. The inertial acceleration felt by the walker is reported in Fig. 14 (centrifugal and Coriolis accelerations are obviously not present, since the platform cannot rotate in this case), while Fig. 15 displays the angular acceleration η_{cont}, directed along the vertical axis, due to a one-foot contact.

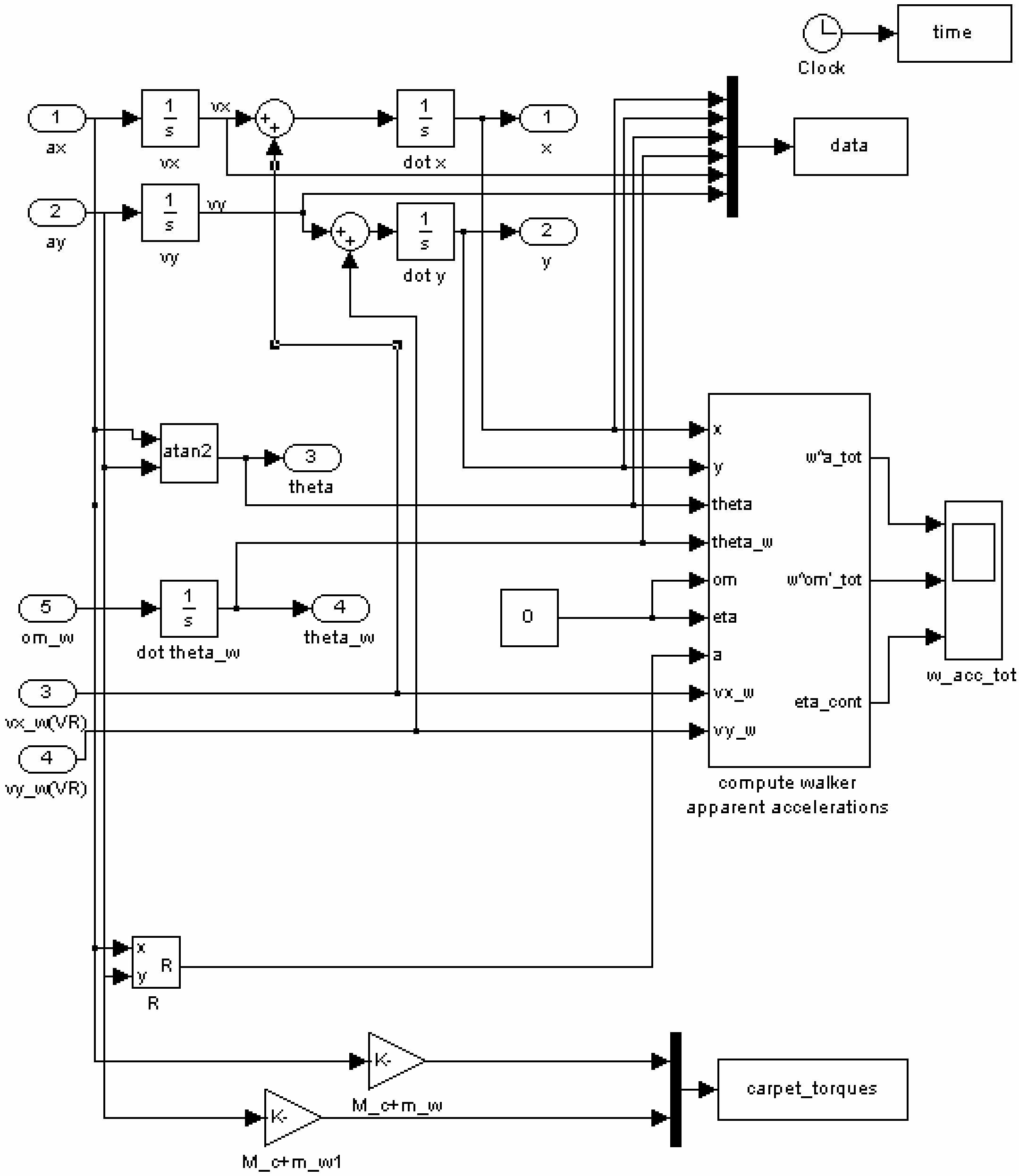

Fig. 13: Omni-directional design: Simulink model for the overall carpet/walker system

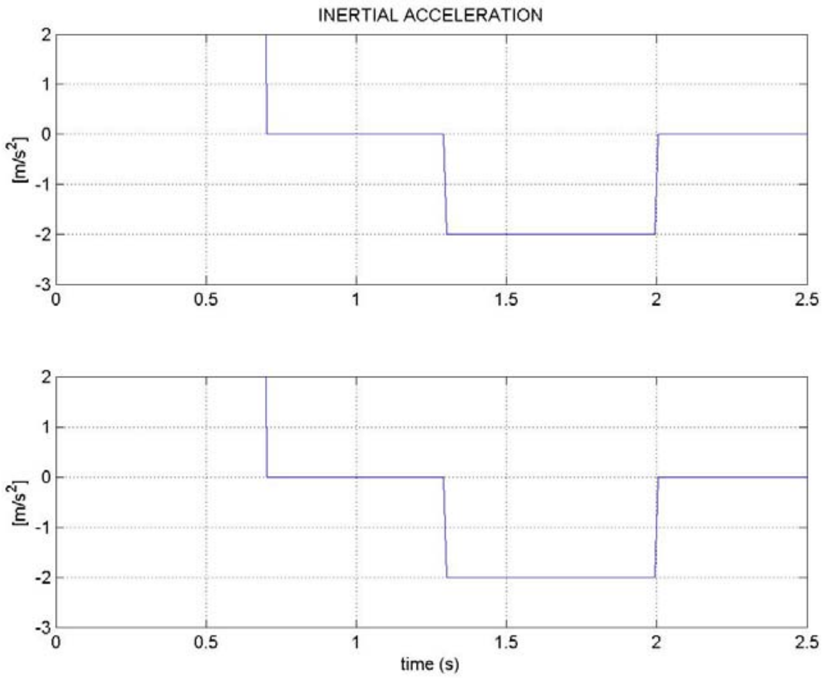

Fig. 14: Omni-directional design: Inertial acceleration felt by the walker in the X_w (top) and Y_w (bottom) directions

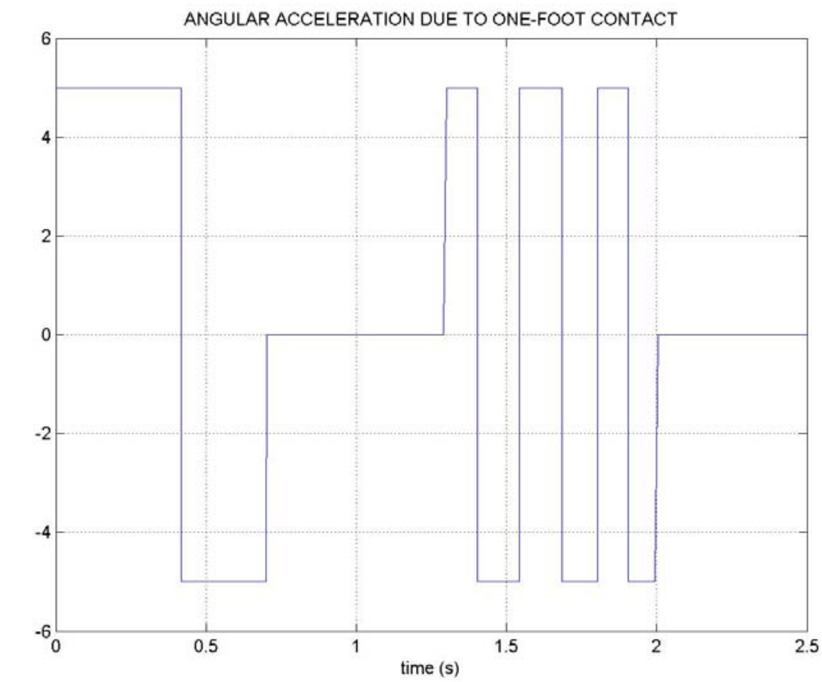

Fig. 15: Omni-directional design: Angular acceleration η_{cont} felt by the walker along the vertical axis, due to one-foot contact

Appendix A: User manual of the BallArrayModel.mdl module

Inputs

- a: linear carpet acceleration;
- eta: angular acceleration of the turntable;
- vx_w(VR): walker velocity in VR (x component of ${}^{t}V_w$, see eq. (5));
- vy_w(VR): walker velocity in VR (y component of ${}^{t}V_w$, see eq. (5));
- om_w: walker angular velocity in VR;

Outputs

- x: absolute walker position (x component);
- y: absolute walker position (y component);
- theta: carpet orientation;
- theta_w: absolute walker orientation;
- v: carpet linear velocity;
- omega: angular velocity of the turntable;

Parameters (defined in InitBallArrayModel.m)

- x0: initial walker position (x component);
- y0: initial walker position (y component);
- theta0: initial carpet orientation;
- theta_w0: initial walker orientation;
- h_w: walker heigh;
- m_w: walker mass;
- I_w: walker inertia;
- d: distance between foot contact point and barycentral vertical axis;
- M_c: carpet mass;
- I_c: carpet/turntable inertia;

Variables created in workspace

- time (1 col): time values;
- data (6 cols): output values;
- w_vel (3 cols): absolute linear and angular walker velocities, expressed in the walker frame;
- w_vel_carpet (3 cols): contribution of carpet motion to w_vel_carpet;
- w_a_inert (4 cols): inertial acceleration felt by the walker, x,y linear comp. + x,y angular comp.;
- w_a_centr (4 cols): centrifugal acceleration felt by the walker, x,y linear comp. + x,y angular comp.;
- w_a_coriolis (4 cols): Coriolis acceleration felt by the walker, x,y linear comp. + x,y angular comp.;
- w_a_tot (4 cols): total apparent acceleration felt by the walker, x,y linear comp. + x,y angular comp.;
- eta_cont (1 col): Angular acceleration η_{cont} felt along the vertical axis, due to one-foot contact
- carpet_torques (2 cols): force, torque provided by the treadmill actuators.

Appendix B: User manual of the OmniDirectionalModel.mdl module

Inputs

- ax, ay: X, Y carpet accelerations;
- vx_w(VR): walker velocity in VR;
- vy_w(VR): walker velocity in VR;
- om_w: walker angular velocity in VR;

Outputs

- x: absolute walker position (x component);
- y: absolute walker position (y component);
- theta: orientation of carpet acceleration input;
- theta_w: absolute walker orientation;
- vx, vy: carpet X and Y velocity components;

Parameters (defined in InitOmniDirectionalModel.m)

- x0: initial walker position (x component);
- y0: initial walker position (y component);
- theta_w0: initial walker orientation;
- h_w: walker heigh;
- m_w: walker mass;
- I_w: walker inertia;
- d: distance between foot contact point and barycentral vertical axis;
- M_c: carpet mass;
- I_c: carpet/turntable inertia;

Variables created in workspace

- time (1 col): time values;
- data (6 cols): output values;
- w_a_inert (4 cols): inertial acceleration felt by the walker, x,y linear comp. + x,y angular comp.;
- eta_cont (1 col): Angular acceleration η_{cont} felt along the vertical axis, due to one-foot contact
- carpet_torques (2 cols): X and Y forces provided by the treadmill actuators.