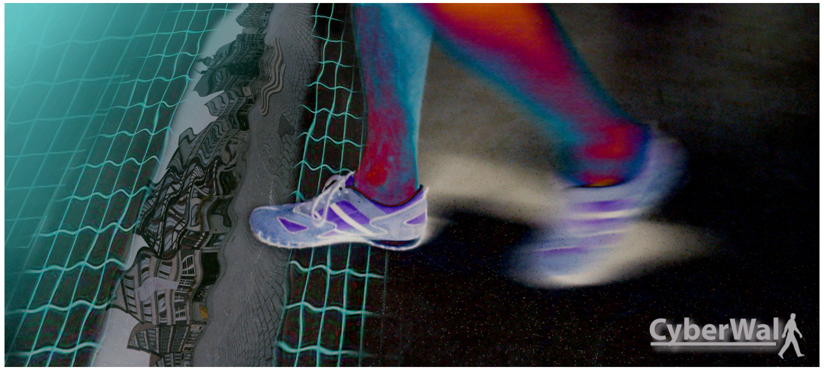
Elective in Robotics

Motion Control of the CyberWalk Platforms – Part I

EU STREP
FP6-511092 project
(2005-2008)



www.cyberwalk-project.org
(no longer active;
see references at the end)

Prof. Alessandro De Luca

Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

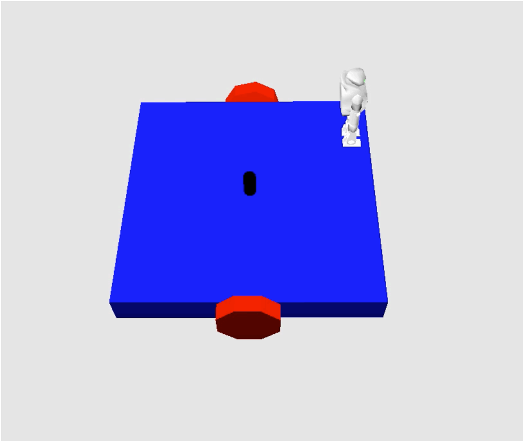
Sapienza Università di Roma

CyberWalk platforms

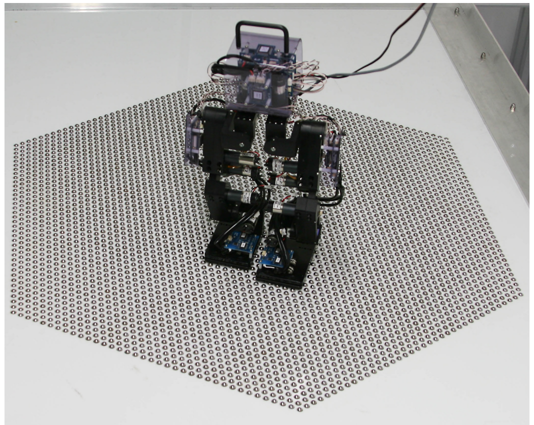
- ball-array/bearing
  - nonholonomic

video

simulation environment

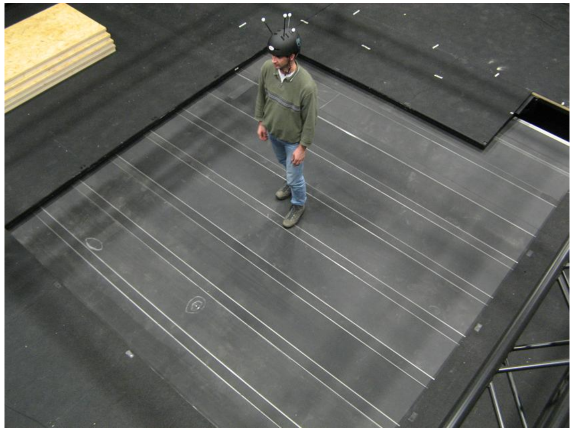


small-scale CyberCarpet



- belt-array
  - omnidirectional

1-D linear treadmill



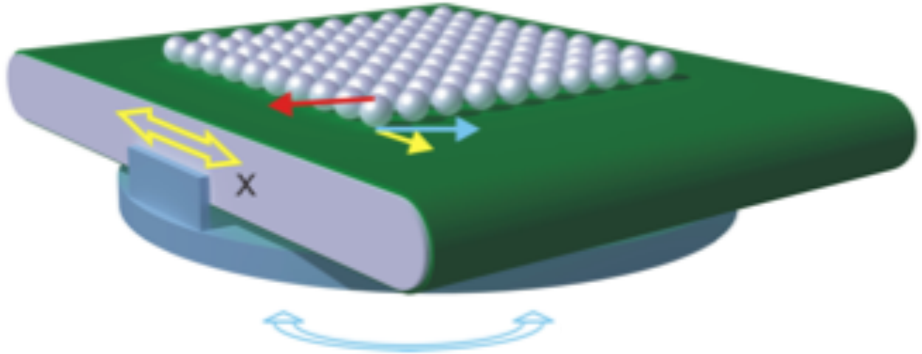
full-scale 2-D platform

Control specifications

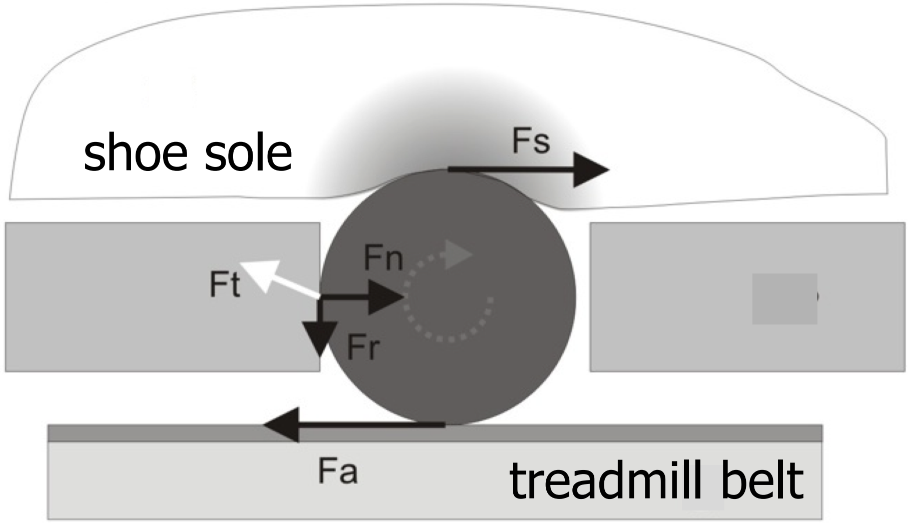
- keep the walker close to the platform center
  - taking into account platform dimensions
  - absolute orientation of walker is not relevant
- satisfy user's perceptual/comfort constraints
  - smoothly controlled motion, especially during start/stop transients
- only measurement of walker position is available
  - visual feedback from external camera system
  - possibly, also information on walker "orientation"
  - intentional walker motion (velocity/acceleration) unknown
- interface/synchronize control commands with VR visualization

CyberCarpet platform
ball-array transmission principle

- a treadmill, mounted on a turntable, with a coverage of the belt by an array of balls arranged in a grid



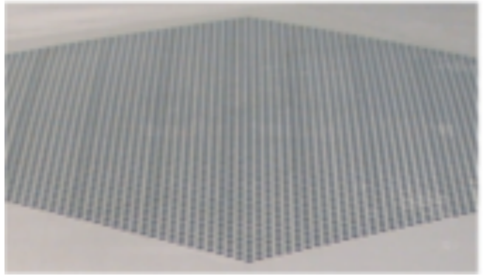
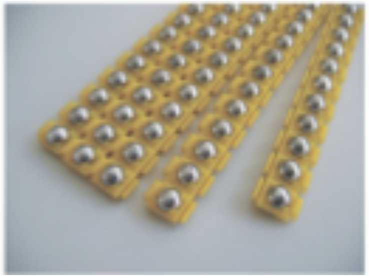
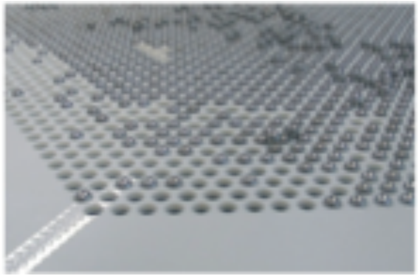
- friction forces acting at the sole-ball, grid-ball, and belt-ball contacts



best friction conditions:
high on sole- and belt-ball,
low on grid-ball contacts

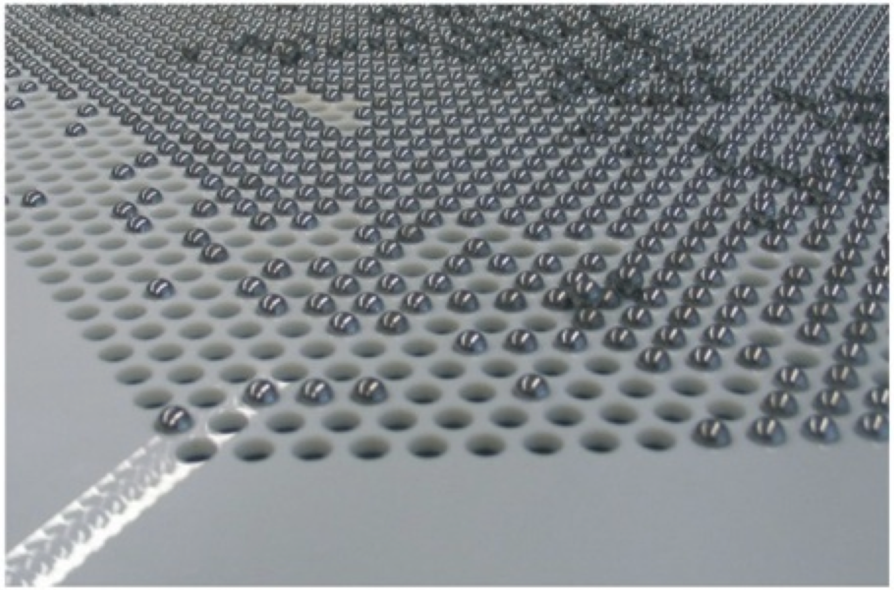
CyberCarpet platform
ball-array and supporting grid

- materials suitable for the grid supporting the balls

INOX steel   Polyethylene (PE) stripes   Acetal (POM) plate

4332 INOX steel balls of d_{ball} = 8 mm diameter with gaps of 0.5 d_{ball} (uniform floor feeling)

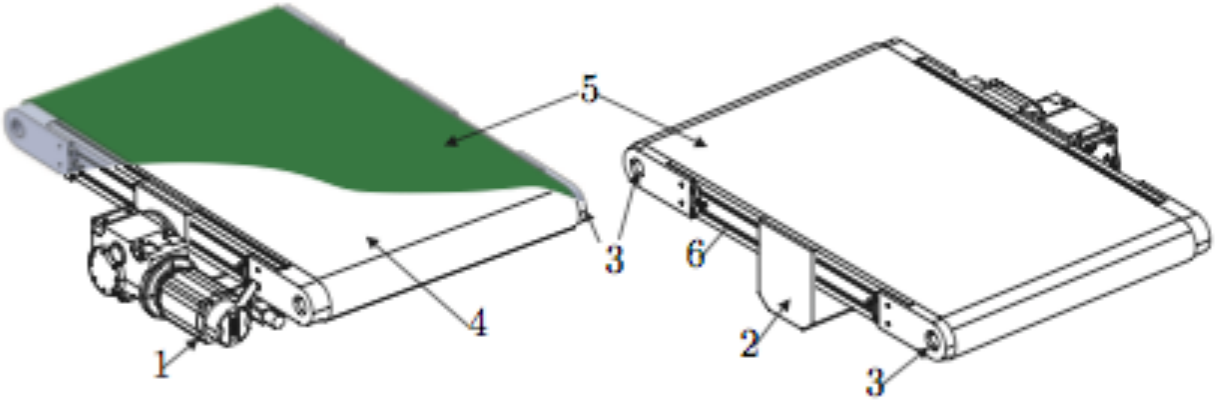


0.3 ball-grid friction coefficient

ball array in a hexagonal grid (inner/outer circles of 683 and 800 mm size)

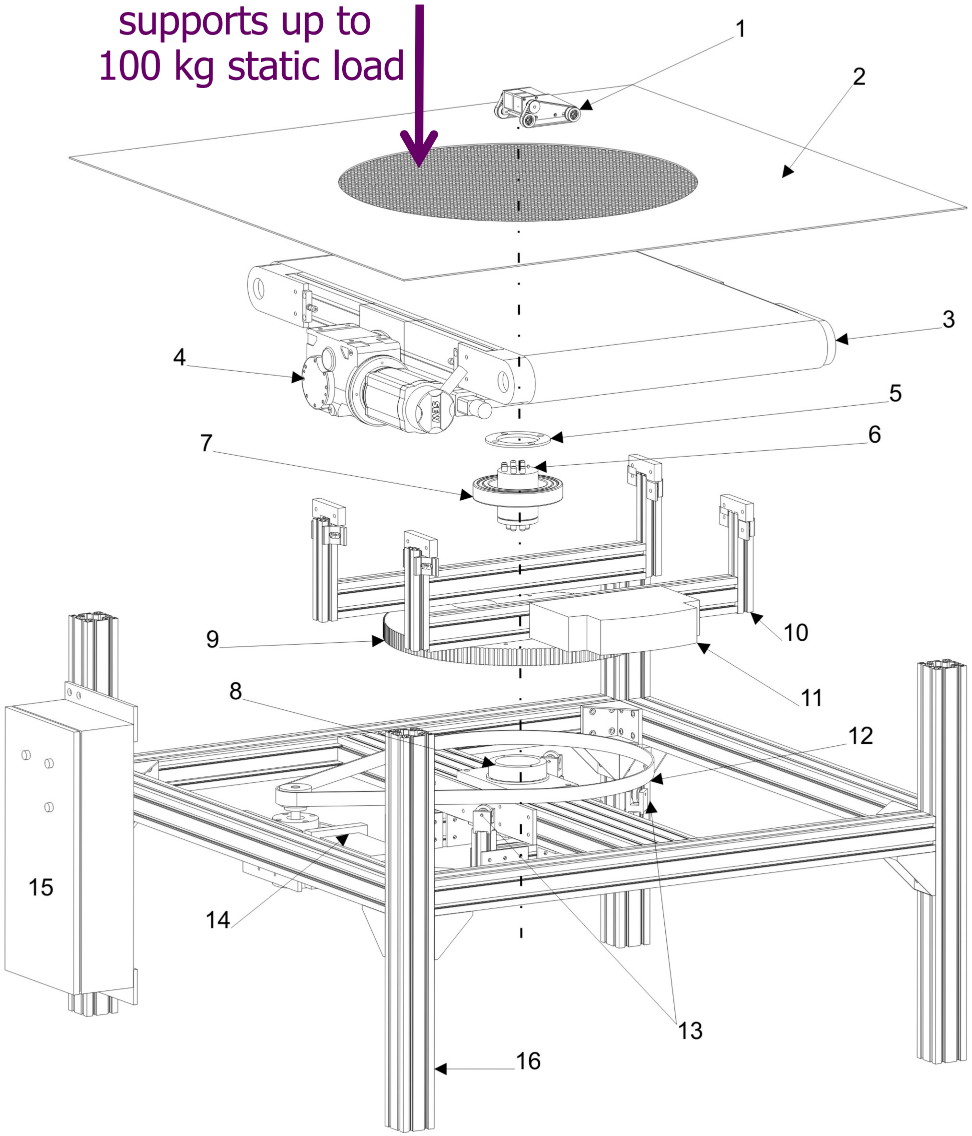
CyberCarpet platform
treadmill, belt and turntable

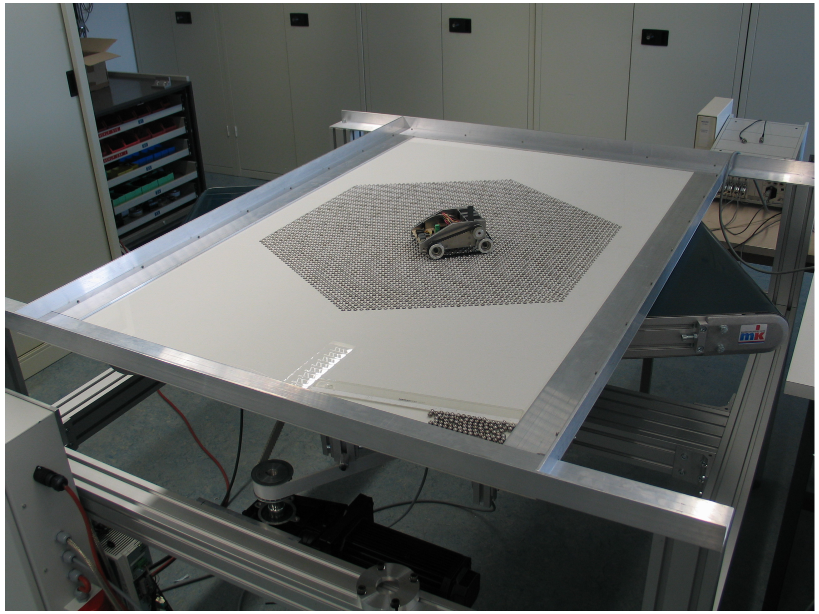
- treadmill
  - linear treadmill of 1.1 m length and 0.8 m width (best ratio ≈ 1.4)
  - Lenze three-phase motor with 2 Nm max torque, ≈ 1:20 transmission ratio
  - max 2 m/s linear speed, max 5 m/s^2 linear acceleration

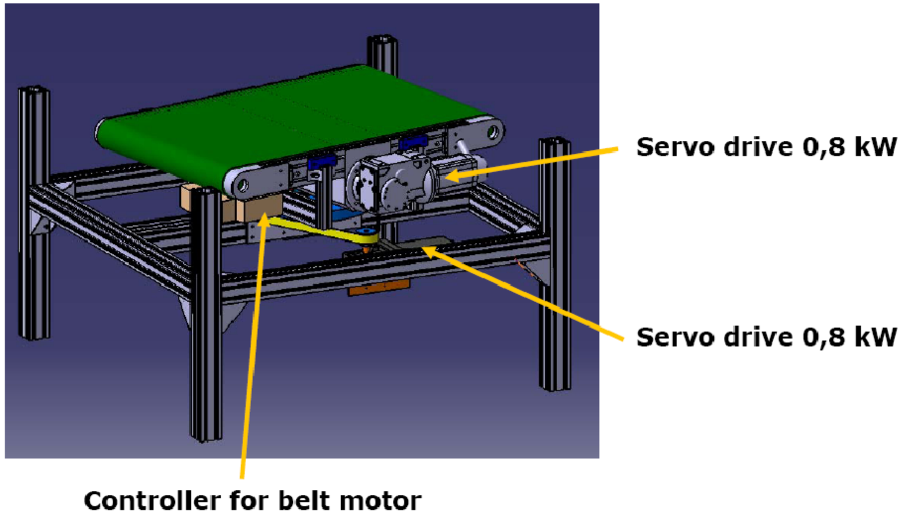


- belt
  - PVC of 5 mm, belt-ball friction coefficient > 0.7
- turntable
  - same Lenze motor ≈ 1:64 transmission ratio (gear + toothed belt)
  - max 2 rad/s angular speed, max 20 rad/s^2 angular acceleration
  - total weight of moving parts (in rotation) ≈ 200 kg

CyberCarpet platform
complete assembly





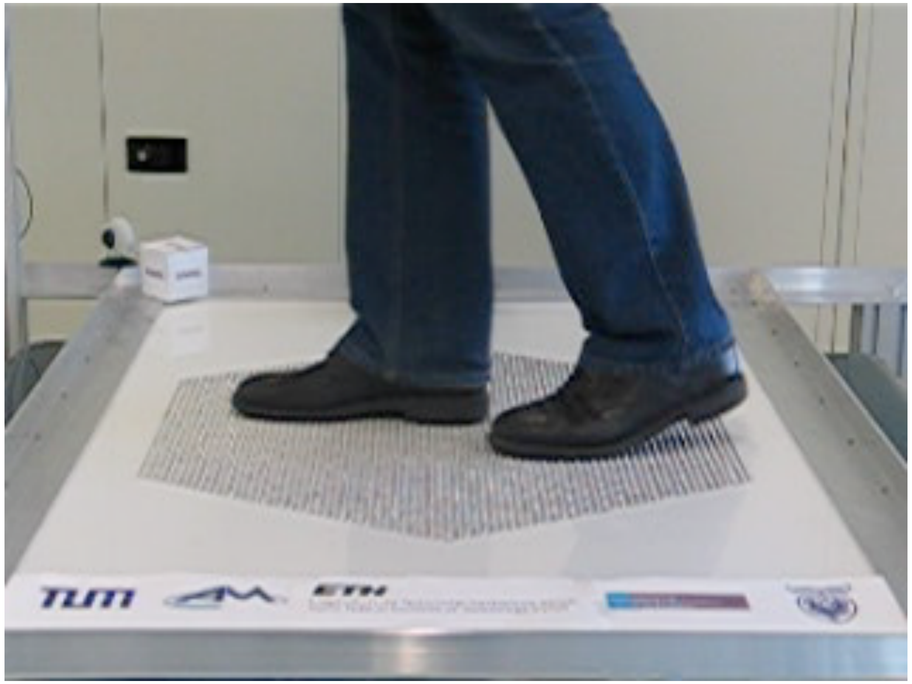


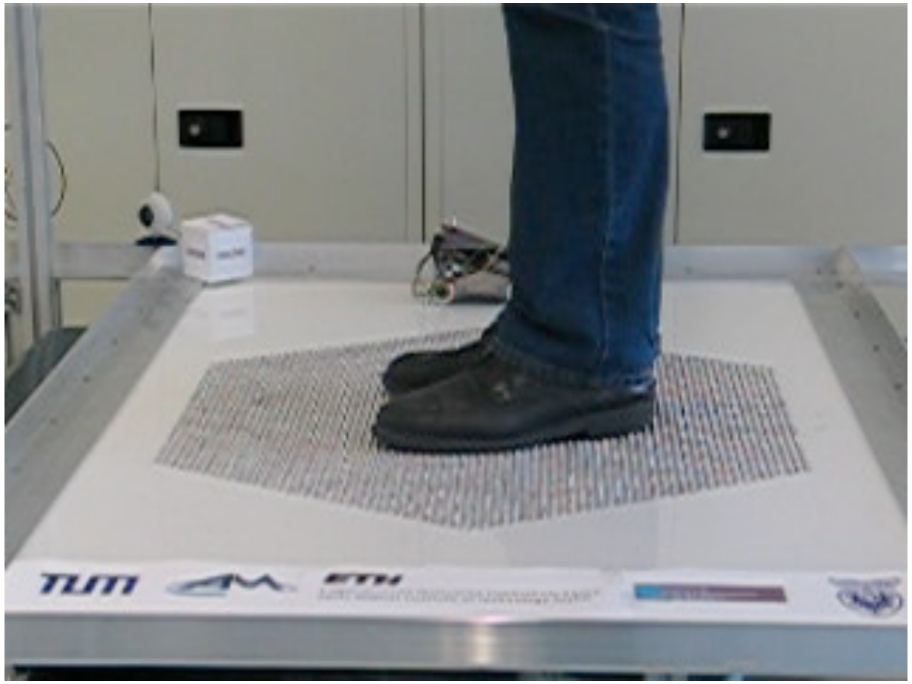
Walking tests
with a human walker

in these tests, no platform control: the walker adapts its speed

video

video

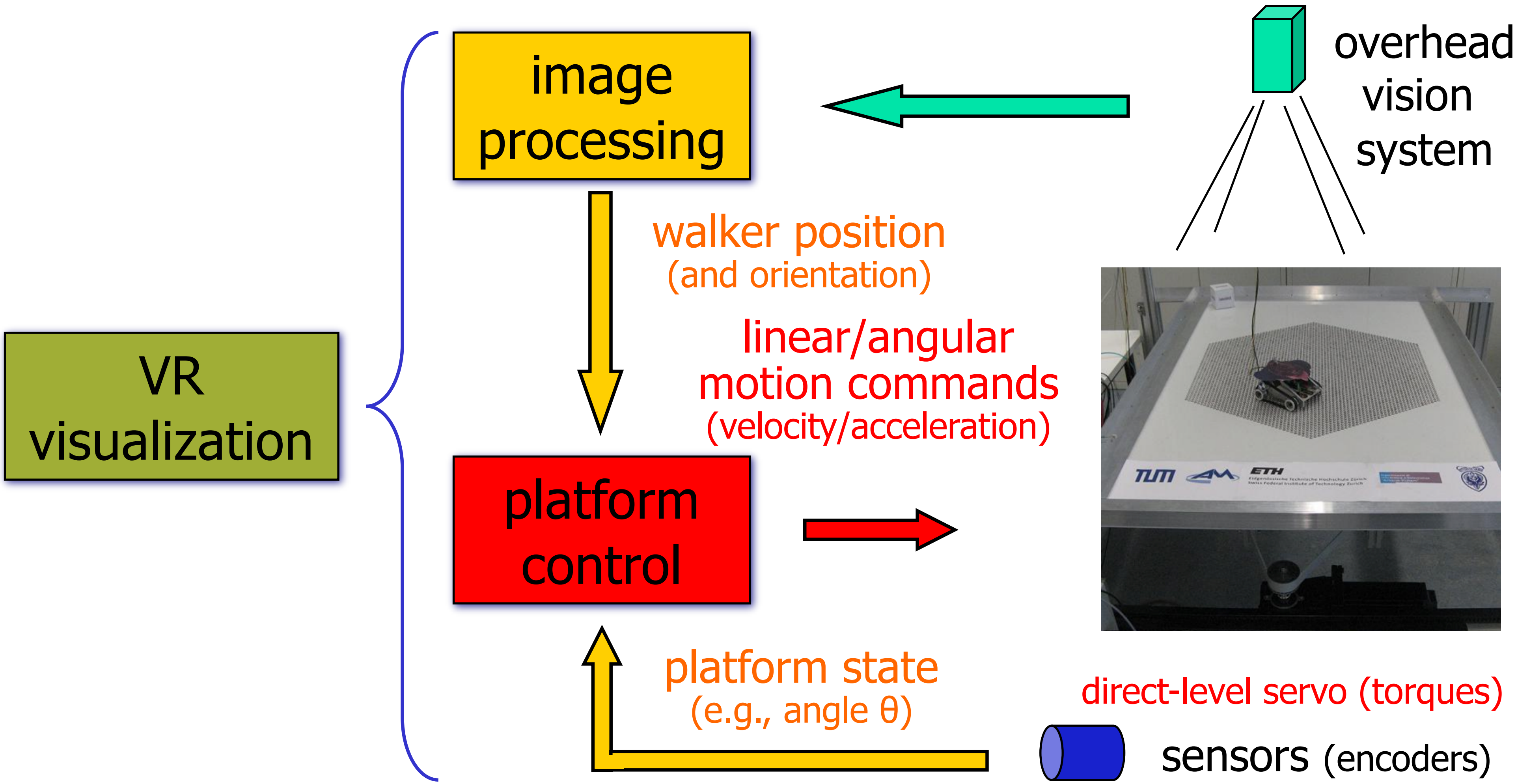




slow speed

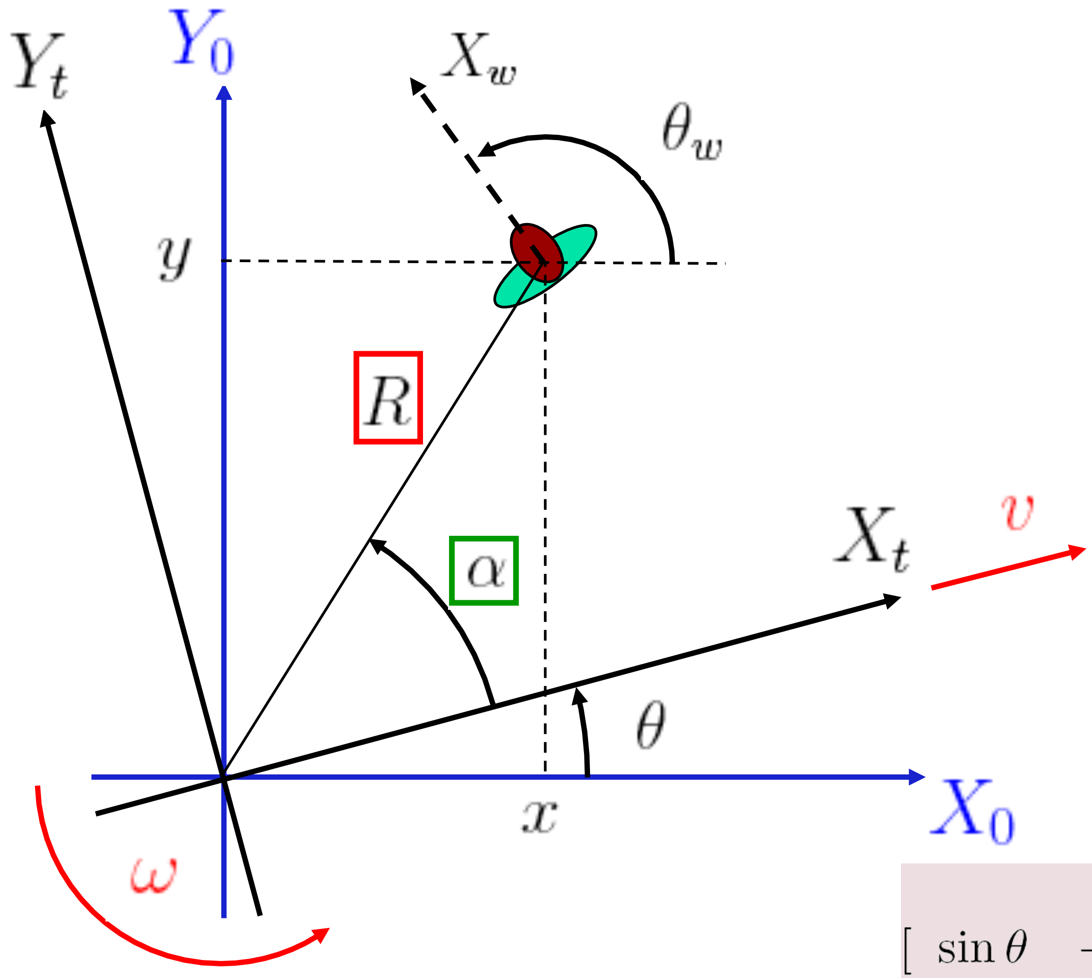
increasing speed
from zero to max
(at 1.4 m/s, balls start jumping out of place
due to dynamic and friction effects with the sole)

System architecture
CyberCarpet platform



Kinematic model

ball-array platform (walker standing still)



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

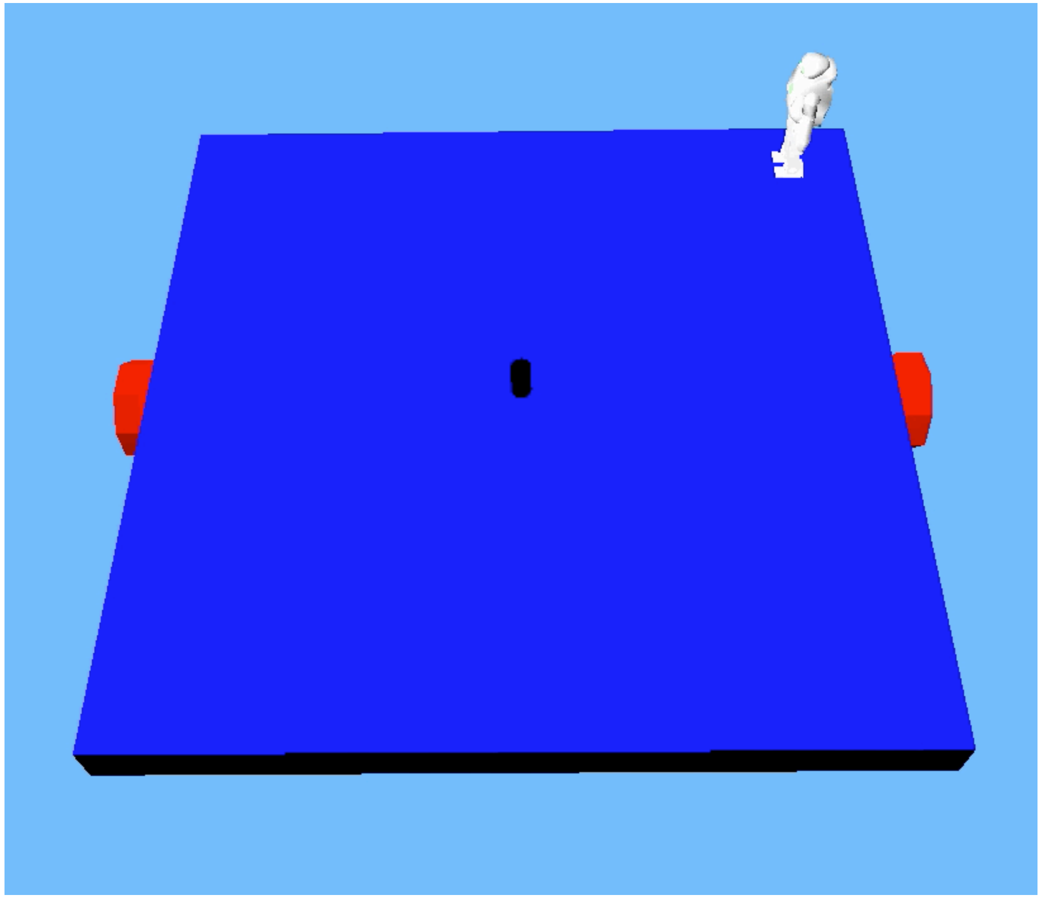
- motion "from below" reversed by balls rolling
- holonomic constraint

$$\theta + \theta_w = const$$

- non-holonomic constraint

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

Holonomic constraint



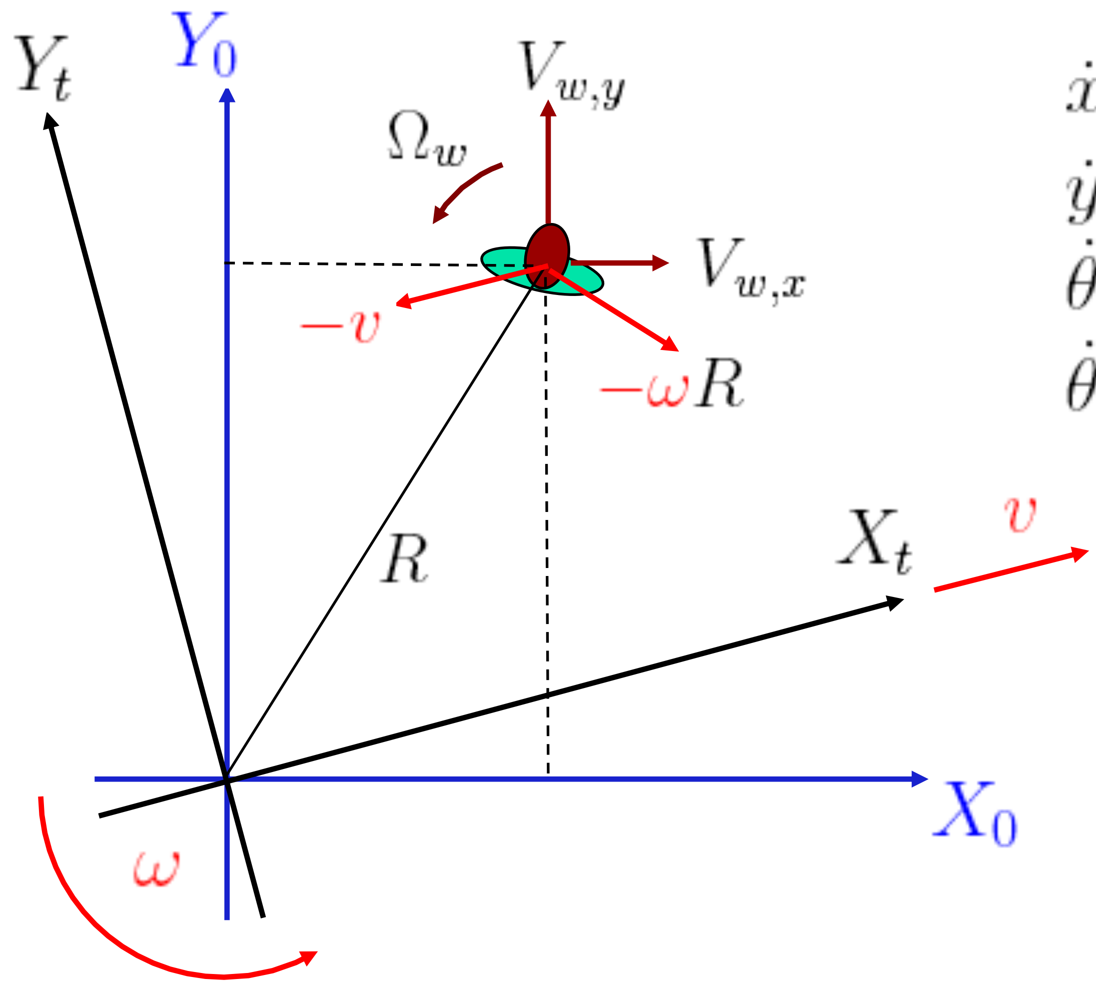
video

one of these two variables
cannot be controlled independently
and is removed from the control problem

$$\theta + \theta_w = \theta(0) + \theta_w(0)$$

Kinematic model

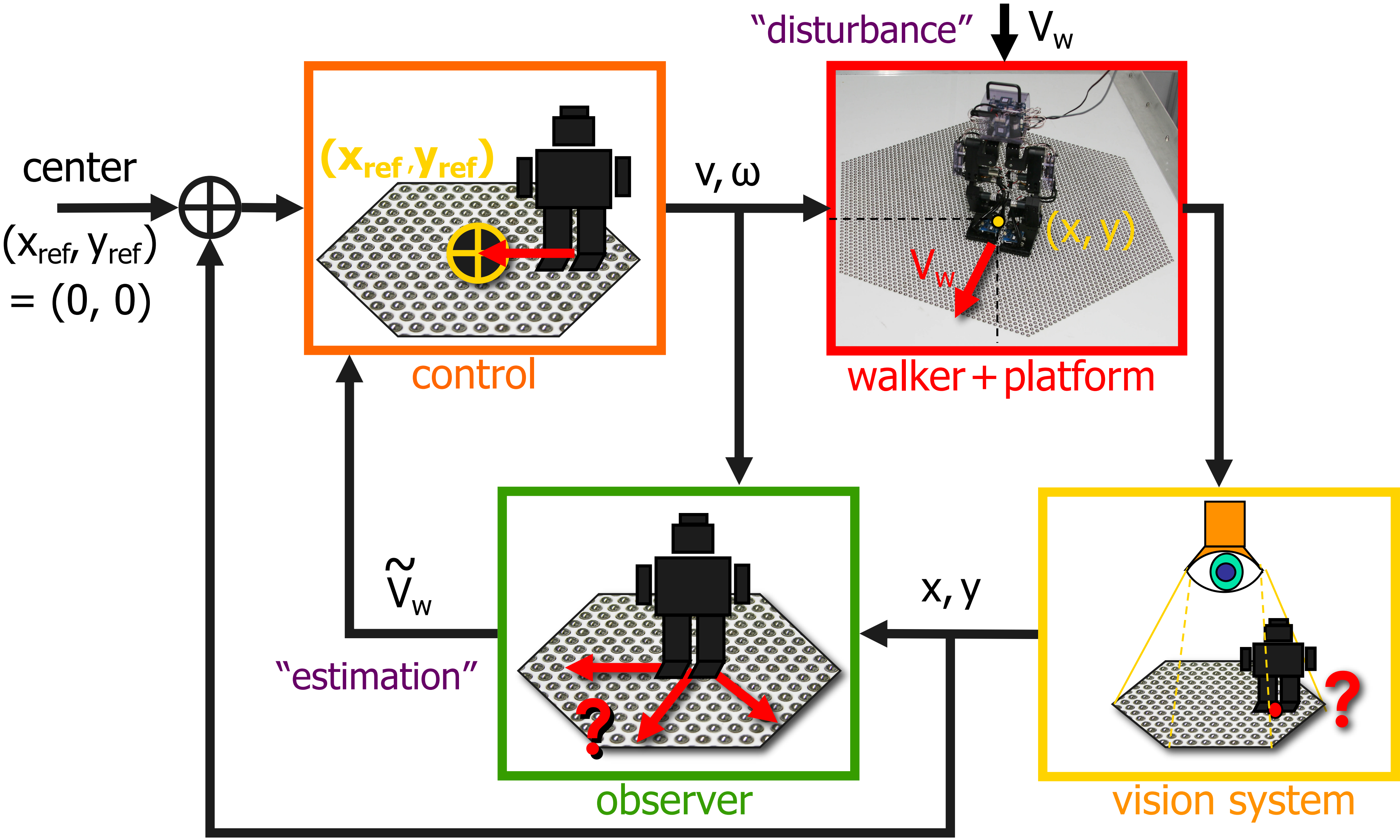
ball-array platform (walker in motion)



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

- walker motion is an unknown "disturbance" for the controller

Control principle
CyberCarpet platform



Control design - 1
CyberCarpet platform

note first that in place of the two Cartesian coordinates (x, y) one can also use the polar coordinates

$$R = \sqrt{x^2 + y^2} \qquad \alpha = \text{ATAN2}(y, x) - \theta$$

with the inverse transformation being

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \text{Rot}(\theta)\, R \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}$$

from this we obtain also $\begin{bmatrix} x\cos\theta + y\sin\theta \\ y\cos\theta - x\sin\theta \end{bmatrix} = \begin{bmatrix} R\cos\alpha \\ R\sin\alpha \end{bmatrix}$

taking the time derivative of polar coordinates (ATAN2 can be replaced by arctan) and substituting, we obtain two equations that could replace the first two in the model (for a standing user)

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

Control design – 1bis

Model derivation using polar coordinates

$$R = \sqrt{x^2 + y^2}$$

$$\dot{R} = \frac{1}{2}\frac{2x\dot{x} + 2y\dot{y}}{\sqrt{x^2 + y^2}} = \frac{x(-v\cos\theta + y\omega) + y(-v\sin\theta - x\omega)}{R}$$

$$= -\frac{v\,(x\cos\theta + y\sin\theta)}{R} = -\frac{v\,R\cos\alpha}{R} = -v\cos\alpha$$

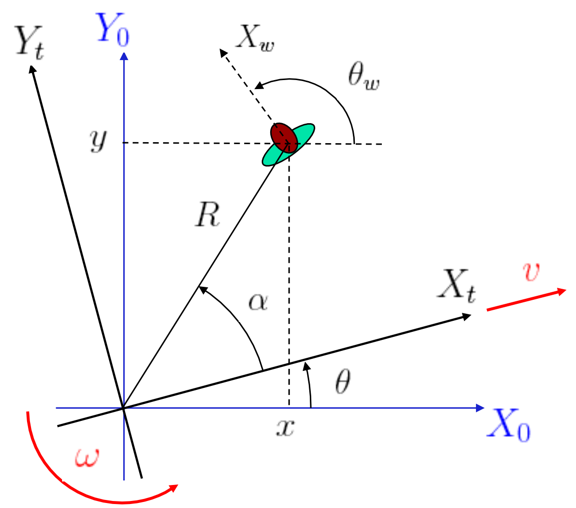
$$\alpha = \arctan\left(\frac{y}{x}\right) - \theta$$

$$\dot{\alpha} = \frac{1}{1 + \left({}^{y}/_{x}\right)^2}\left(\frac{\dot{y}}{x} - \frac{y\dot{x}}{x^2}\right) - \dot{\theta} = \frac{1}{x^2 + y^2}(\dot{y}x - y\dot{x}) - \omega$$

$$= \frac{1}{R^2}[x(-v\sin\theta - x\omega) - y(-v\cos\theta + y\omega)] - \omega$$

$$= \frac{1}{R^2}[v\,(y\cos\theta - x\sin\theta) - \omega\,(x^2 + y^2)] - \omega = \frac{1}{R^2}[v\,R\sin\alpha - \omega\,R^2] - \omega$$

$$= v\,\frac{\sin\alpha}{R} - 2\omega$$



Control design - 2
CyberCarpet platform

- consider first the case of a user standing still: $V_w = 0 \quad \Omega_w = 0$
- we are interested in controlling (x, y) only, and we will do this by an input-output feedback linearization method; it is

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

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

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

obtaining linear, decoupled behavior in terms of the new commands v_1, v_2

- choosing then $v_1 = -k_1 x, \; v_2 = -k_2 y$ with positive gains $k_i, i = 1{,}2$, the user is exponentially stabilized to the origin

$$x(t) = e^{-k_1 t} x_0 \qquad y(t) = e^{-k_2 t} y_0$$

Control design - 3
CyberCarpet platform

- setting (from now on) $k_1 = k_2 = k > 0$, it follows that

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

"straight line" recover to the center
if the walker is standing still

- the resulting control law (written in either set of coordinates) is

(1)
$$\begin{cases} v = \dfrac{k(x^2+y^2)}{x\cos\theta + y\sin\theta} = \dfrac{kR^2}{R\cos\alpha} = \dfrac{kR}{\cos\alpha} \\ \omega = \dfrac{k(y\cos\theta - x\sin\theta)}{x\cos\theta + y\sin\theta} = \dfrac{kR\sin\alpha}{R\cos\alpha} = k\tan\alpha \end{cases}$$

which is (as expected!) singular when

$\alpha = \pm\frac{\pi}{2}$ or/and $R = 0$ (α is not defined)

namely for $R\cos\alpha = x\cos\theta + y\sin\theta = 0$

- the control law should be modified so as to handle these singularities

Control design - 4
CyberCarpet platform

- when $R \neq 0$, singularity at $\alpha = \pm\frac{\pi}{2}$ is eliminated multiplying (1) by $|\cos\alpha|$

(2) $$\begin{cases} v = kR\,\mathrm{sgn}(\cos\alpha) \\ \omega = k\sin\alpha\,\mathrm{sgn}(\cos\alpha) \end{cases}$$ with $\mathrm{sgn}(arg) = \begin{cases} +1 & \text{for } arg \geq 0 \\ -1 & \text{for } arg < 0 \end{cases}$

- the output dynamics of the controlled system is no longer linear

$$\dot{x} = -k\,|\cos\alpha|\,x, \qquad \dot{y} = -k\,|\cos\alpha|\,y$$

and closed-loop asymptotic stability requires a Lyapunov/LaSalle analysis

- let $\theta + \alpha = \theta_0 + \alpha_0 =: \beta_0$ be the constant angle pointing to the walker; the Lyapunov candidate and its time derivative are

$$V(x,y,\theta) = \tfrac{1}{2}(x^2 + y^2 + \sin^2(\beta_0 - \theta)) = \tfrac{1}{2}(R^2 + \sin^2\alpha) \geq 0,$$

$V = 0$ iff $(x,y,\theta) \in \mathcal{S}$ with $\mathcal{S} = \{(0,0,\theta) : \sin(\beta_0 - \theta) = 0\}$

➡ $$\dot{V} = R\dot{R} + \sin\alpha\,\cos\alpha\,\dot{\alpha} = -k\,|\cos\alpha|\,(R^2 + \sin^2\alpha) \leq 0$$

$\dot{V} = 0$ if $(x,y,\theta) \in \mathcal{S}$ or $\cos\alpha = 0$

➡ at $\cos\alpha = 0$ it is $\omega = \pm k$ (non-invariant state)

➡ $\mathcal{S}$ is asymptotically stable

Control design - 5
CyberCarpet platform

- at $R = 0$, control (2) is not smooth at the origin (chattering problems); multiplying it further by R yields finally

$$
\begin{aligned}
v &= kR^2 \operatorname{sgn}(\cos\alpha) \\
&= k(x^2 + y^2) \operatorname{sgn}(x\cos\theta + y\sin\theta) \\
\omega &= kR\sin\alpha \operatorname{sgn}(\cos\alpha) \\
&= k(y\cos\theta - x\sin\theta) \operatorname{sgn}(x\cos\theta + y\sin\theta)
\end{aligned}
\quad (*)
$$

smooth singularity-free velocity-level feedback law

- asymptotic stability properties can be proved similarly to the previous case by Lyapunov/LaSalle arguments
- "straight line" recover of the standing walker to the center still holds

Control design - 6
Walker in motion on CyberCarpet platform

- when the walker is in motion (with an intentional linear velocity V_w), control (1) or (2) or (*) do not allow the walker to recover the center
- for example, if the walker is moving (in the virtual space) in a straight line with constant speed $\bar{V}$, there will be a steady-state position error with control (1) at a distance $\bar{R} = \bar{V}/k$ from the center
- in this particular case, the addition of an integral action in the auxiliary commands

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

  will zero the steady-state error
- there is, however, a problem of overshooting; in addition, for more general cases, there is no guarantee that this will work
- a disturbance observer can be set up to estimate the intentional linear velocity of the walker, so as to use it for control purposes

Control design - 7
Estimating walker velocity on CyberCarpet platform

- an estimate $\tilde{V}_w$ of the walker intentional linear velocity V_w is obtained by two dynamic observers (one for each component) with states ξ_x and ξ_y

$$\dot{\xi}_x = -v\cos\theta + y\omega + k_w(x - \xi_x)$$
$$\tilde{V}_{w,x} = k_w(x - \xi_x)$$
$$\dot{\xi}_y = -v\sin\theta - x\omega + k_w(y - \xi_y)$$
$$\tilde{V}_{w,y} = k_w(y - \xi_y)$$

$$k_w > 0$$

the actual commands sent to the platform should be used here

which are copies of the system dynamics, with a forcing term in place of the unknown walker velocity estimate, and where only measured positions and input commands are used

- it is easy to verify that the estimate is a first-order stable filter of the intentional walker's linear velocity

$$\dot{\tilde{V}}_{w,x} = k_w\left(V_{w,x} - \tilde{V}_{w,x}\right)$$
$$\dot{\tilde{V}}_{w,y} = k_w\left(V_{w,y} - \tilde{V}_{w,y}\right)$$

in Laplace domain

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

e.g., for constant walker velocity, the estimation error goes to zero

Final feedback/feedforward control
CyberCarpet platform

- the estimate is used within a feedforward term, added to the previous feedback centering term, to counteract the “disturbance” V_w

$$v = v_{fb} + v_{ff} = v_{fb} + \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \tilde{V}_w$$

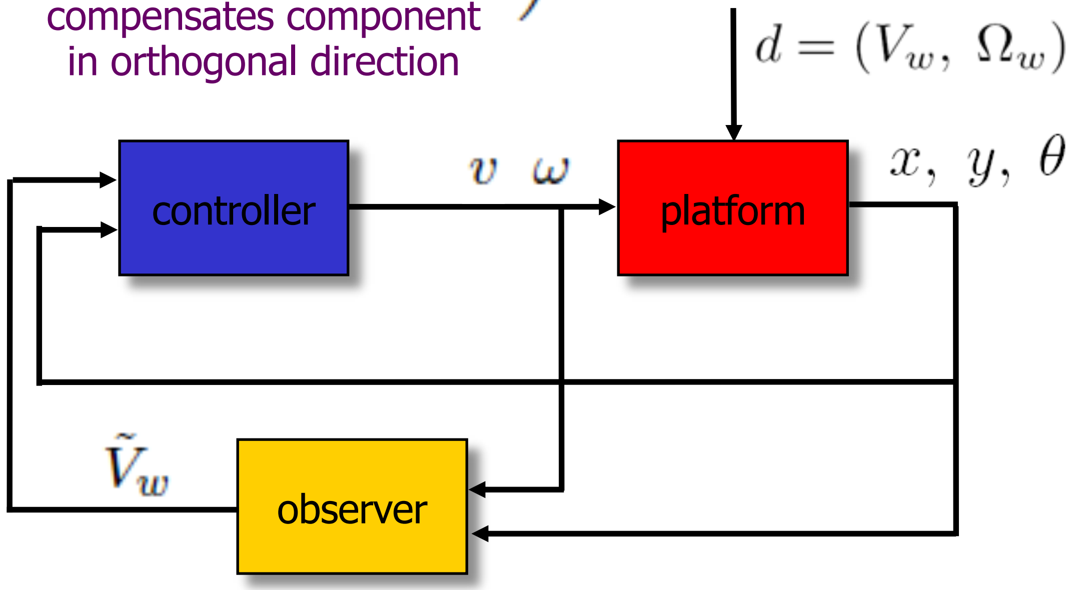
feedback stabilizing parts

compensates component in treadmill direction

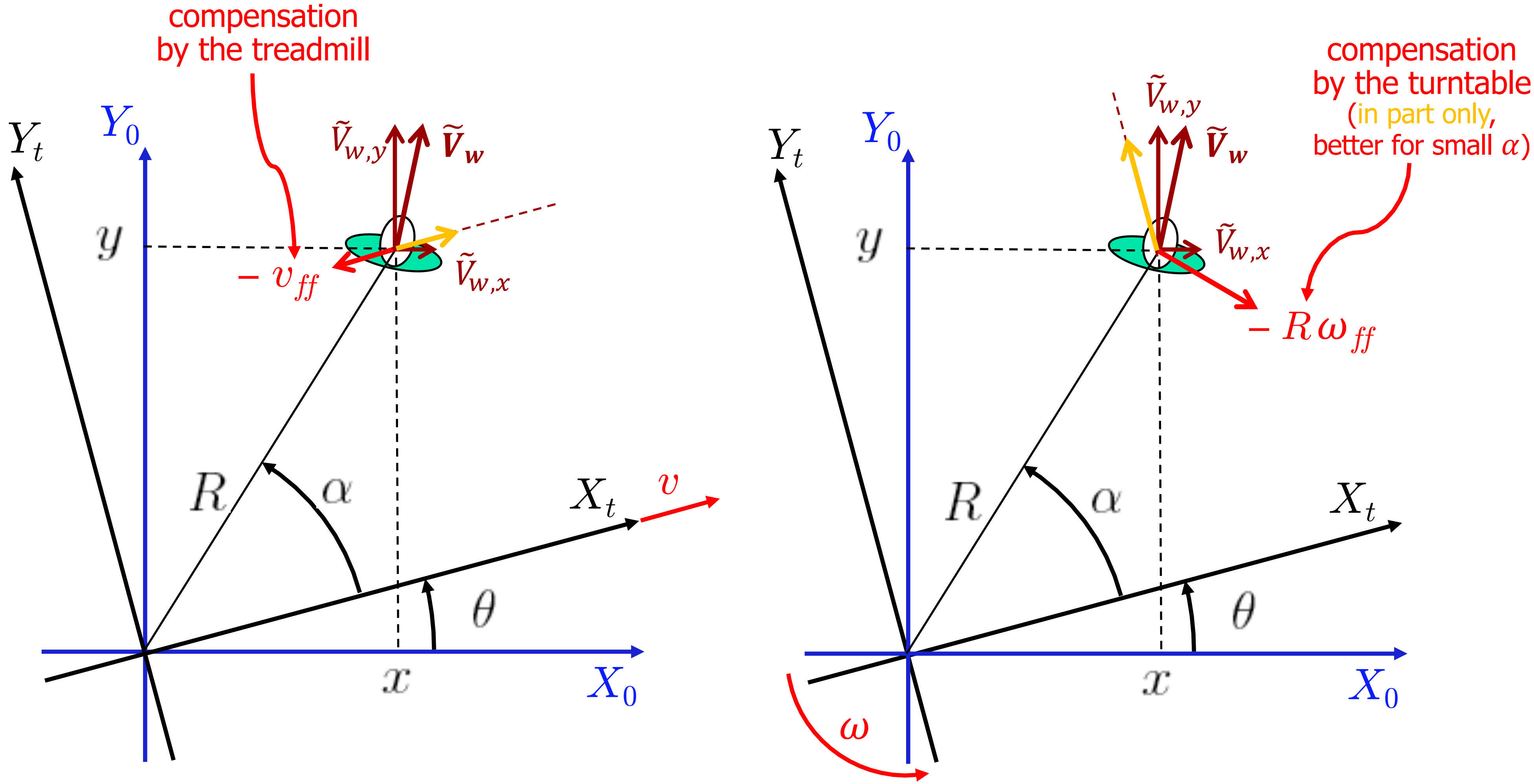
$$\omega = \omega_{fb} + \omega_{ff}$$
$$= \omega_{fb} + \text{sat}\left(\frac{1}{R}\begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix} \tilde{V}_w\right)$$

compensates component in orthogonal direction

saturates to $\pm\omega_{ff,max}$ for $R \longrightarrow 0$



Role of feedforward actions
CyberCarpet platform



Comments on the final control law
CyberCarpet platform

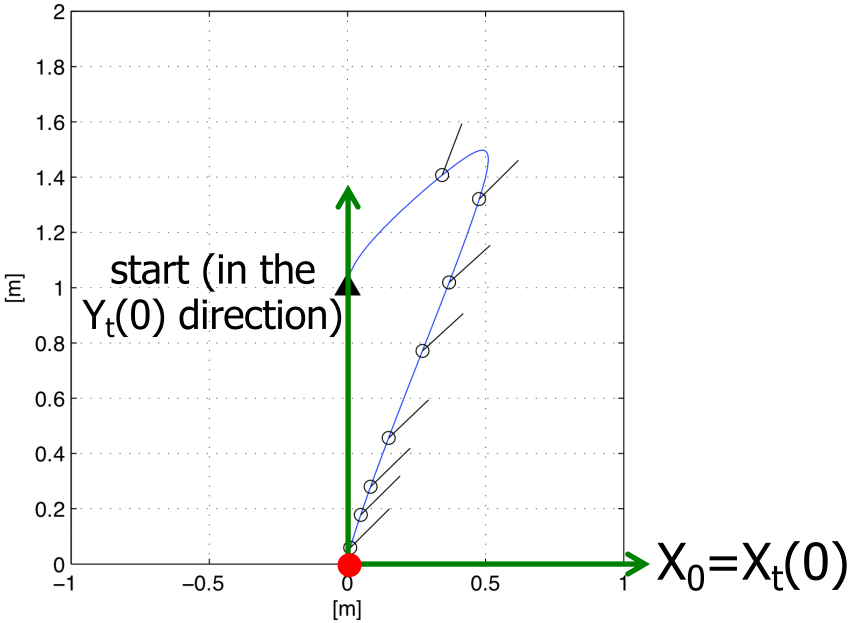
- including intentional angular velocity in the disturbance observer
  - feasible, but not done here ...
- gain scaling of feedback part, so as to satisfy perceptual constraints

$$|v| \leq v_{max} \quad |\omega| \leq \omega_{max} \quad \Rightarrow \quad k = \frac{\bar{k}}{\max\left\{1, \frac{|v|}{v_{max}}, \frac{|\omega|}{\omega_{max}}\right\}} > 0$$

  - the feedforward action needs not to be scaled since it contributes to canceling absolute motion of the walker, thus reducing perceptual effects
- saturation in the feedforward angular velocity
  - avoids control explosion for R approaching zero
  - for sufficiently smooth intentional velocity of the walker, the platform tends to align with V_w so that the ω_{ff} remains anyway small

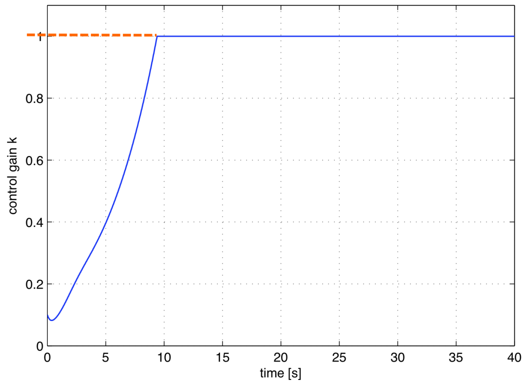
Simulations with the final control law

straight line walk at 1 m/s



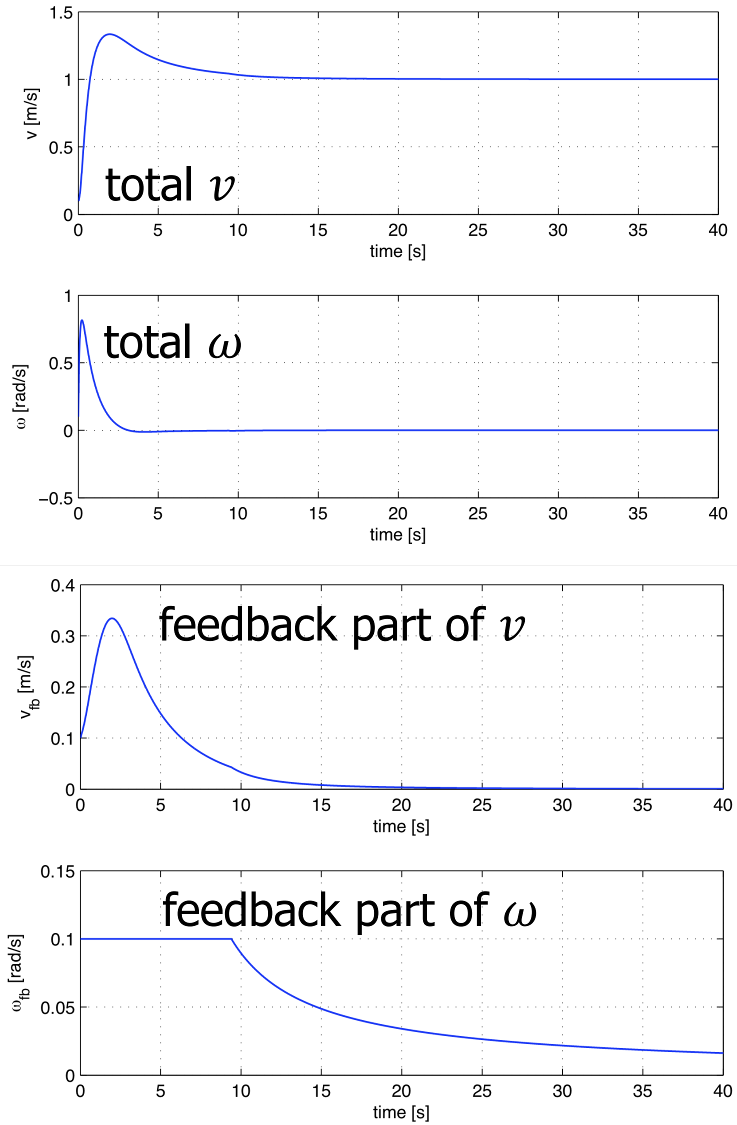
absolute trajectory

$\bar{k} = 1$



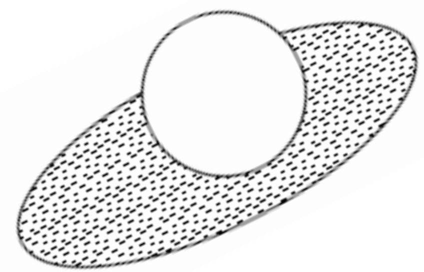
feedback gain scaling for
v_{max} = 0.5 m/s
ω_{max} = 0.1 rad/s

$k_w = 10 \quad \omega_{ff,max} = 2$ rad/s



Walker geometric model
for visual tracking (top view)

- shoulders and head of the walker modeled as an ellipsoid and a circle



$$\boldsymbol{s} = \{x, y, \theta_w, c_x, c_y\}$$

- x, y: ellipsoid center
- θ_w: orientation of minor ellipsoid axis
- c_x, c_y: circle center (relative to ellipse)

areas may vary due to changes in human pose, but ellipsoid and circle sizes have been kept constant

- each particle is an hypothesis about the state of the walker, with an associated (Gaussian) probability distribution
- at each iteration, the visual tracker generates a set of new hypotheses propagating particles through a simple dynamics
- this prior distribution of the next state is then tested using the observation of the image captured by the overlooking camera, and probabilities associated
- a set of particles is redrawn, modeling the posterior probability distribution of the walker state, from which the current position/orientation is extracted

Visual tracking algorithm
color-based particle filter

- N particles are equally initialized (mouse clicks on first image)

$$s^{(1)},\ldots, s^{(N)} \qquad \pi^{(j)},\ j = 1,\ldots,N \qquad (N = 500)$$

Algorithmic steps, at each iteration t:

- evolution $s_t = s_{t-1} + w_{t-1}$ ⟵ random Gaussian variable
- color (normalized) histograms p and p' are built for the pixels in the shoulder−head and head regions and compared to stored q and q' histograms using the Bhattacharyya (B.) coefficients $\rho[p,q]$ and $\rho[p',q']$

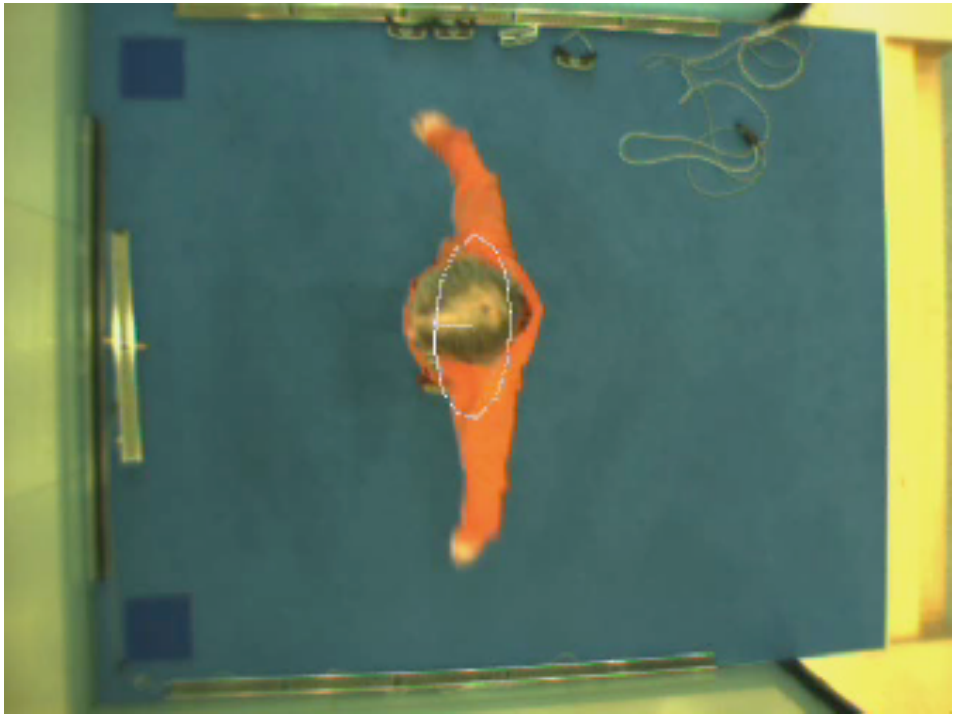
$$d = \sqrt{1 - \frac{\rho[p,q] + \rho[p',q']}{2}} \longrightarrow s^{(j)} \longrightarrow \pi^{(j)} = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{d^2}{2\sigma^2}\right)$$

B. distance = similarity measure — probabilities associated to particles

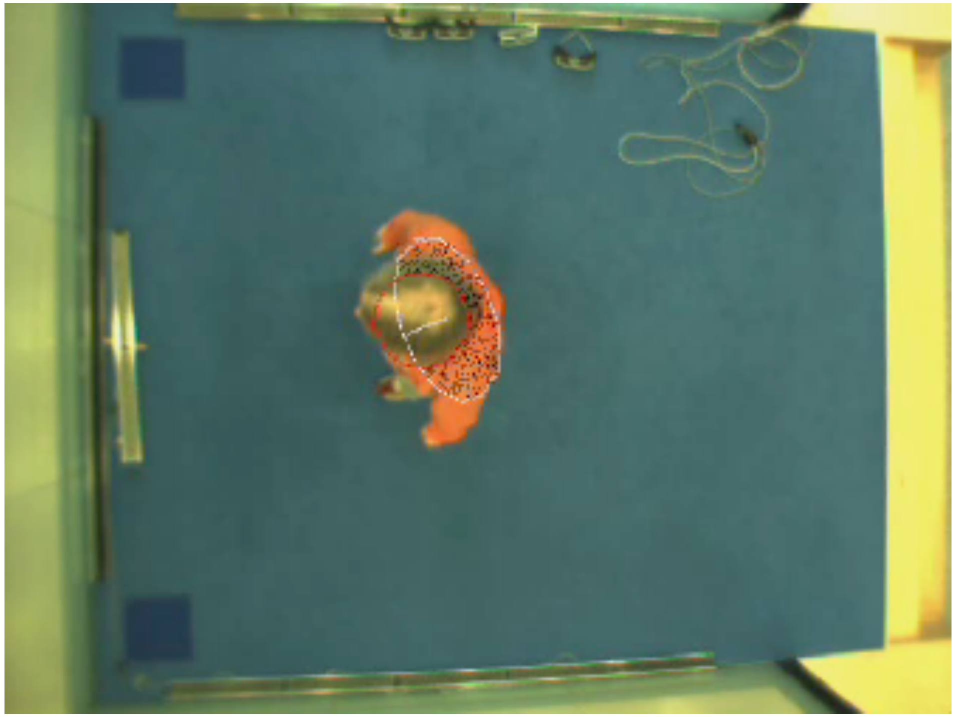
- a set of N new particles is drawn (with replacement) by random sampling using these probability distributions
- the walker's current pose is obtained by weighted averaging of the current particle set, where the weights are the B. distances $d^{(j)}$

Full-scale visual tracking
from single overhead camera

video | video



ellipsoid only

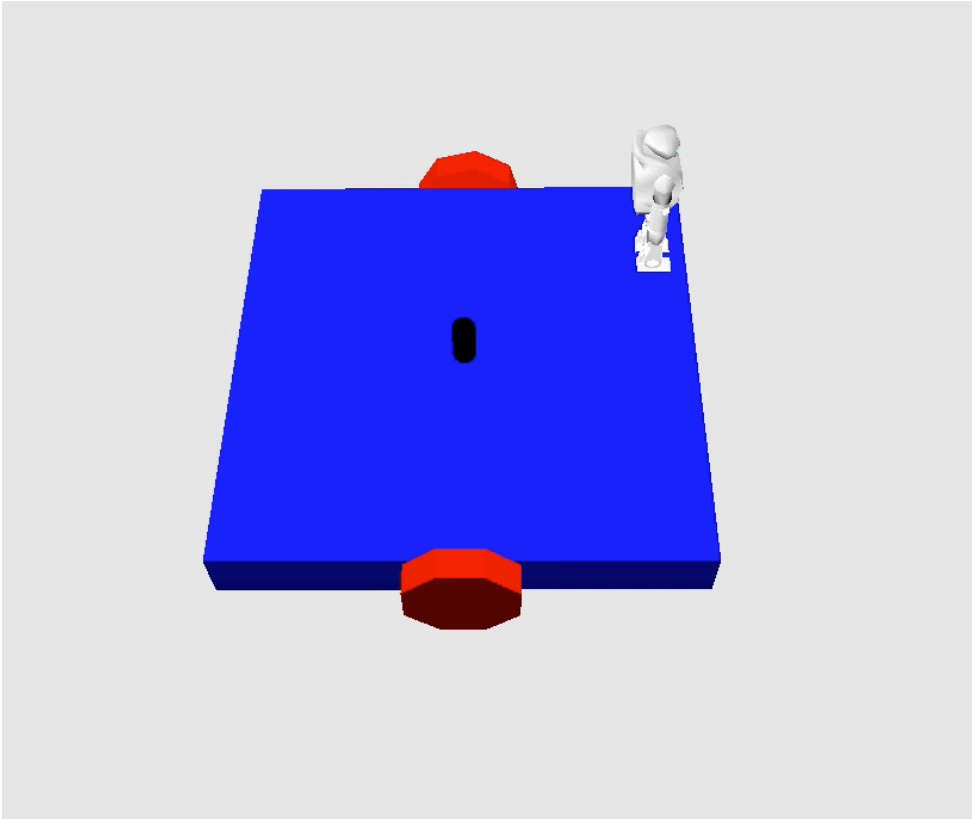


ellipsoid plus head

on-line visual localization of walker position and orientation

3 Hz rate with basic algorithm was improved to 17 Hz, by constructing histograms only for 500 randomly selected points in the relevant regions

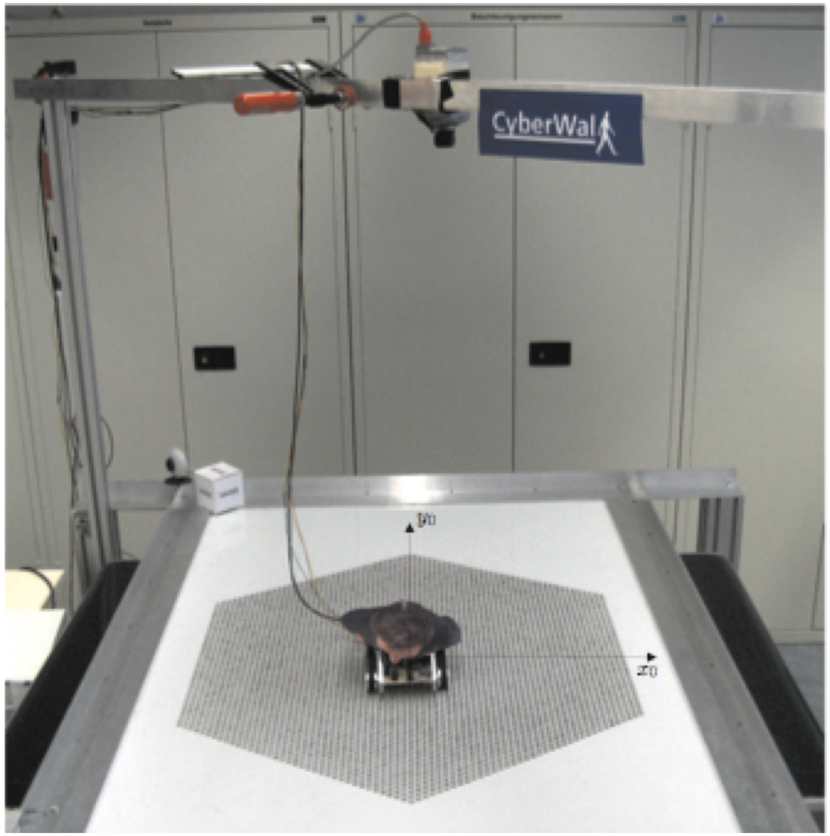
Simulation
CyberCarpet using the visual data



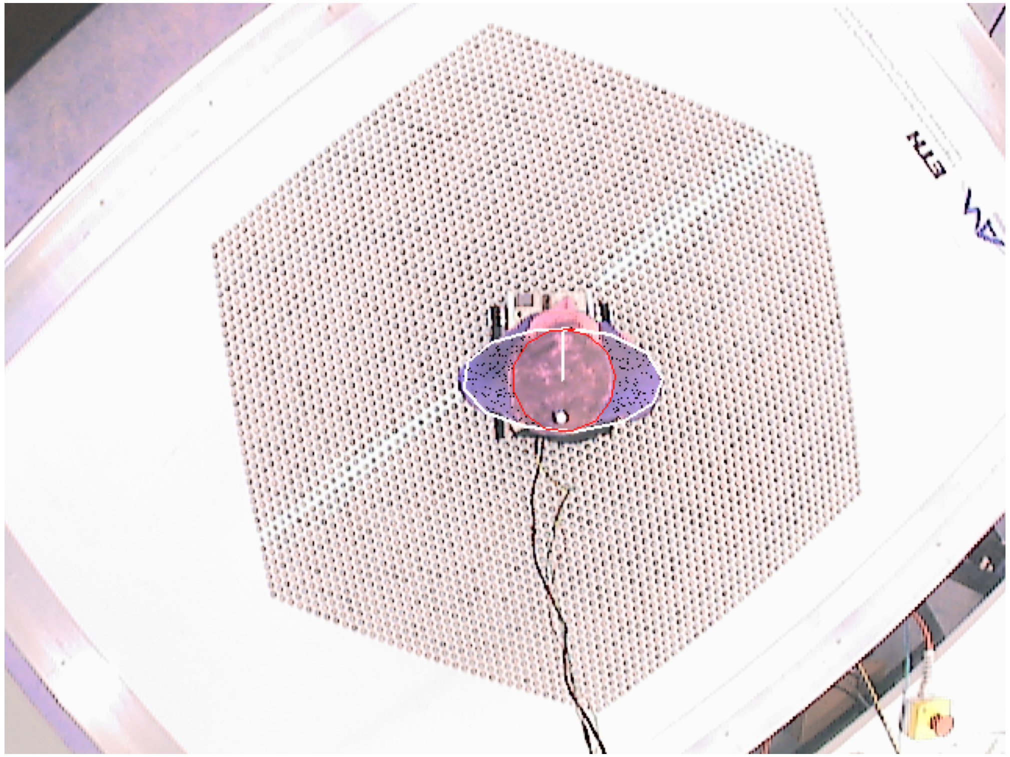
video

ball-array platform velocity controller fed by
the real full-scale walker position (random walk)
obtained with the previous visual tracking system

Actual experimental set-up

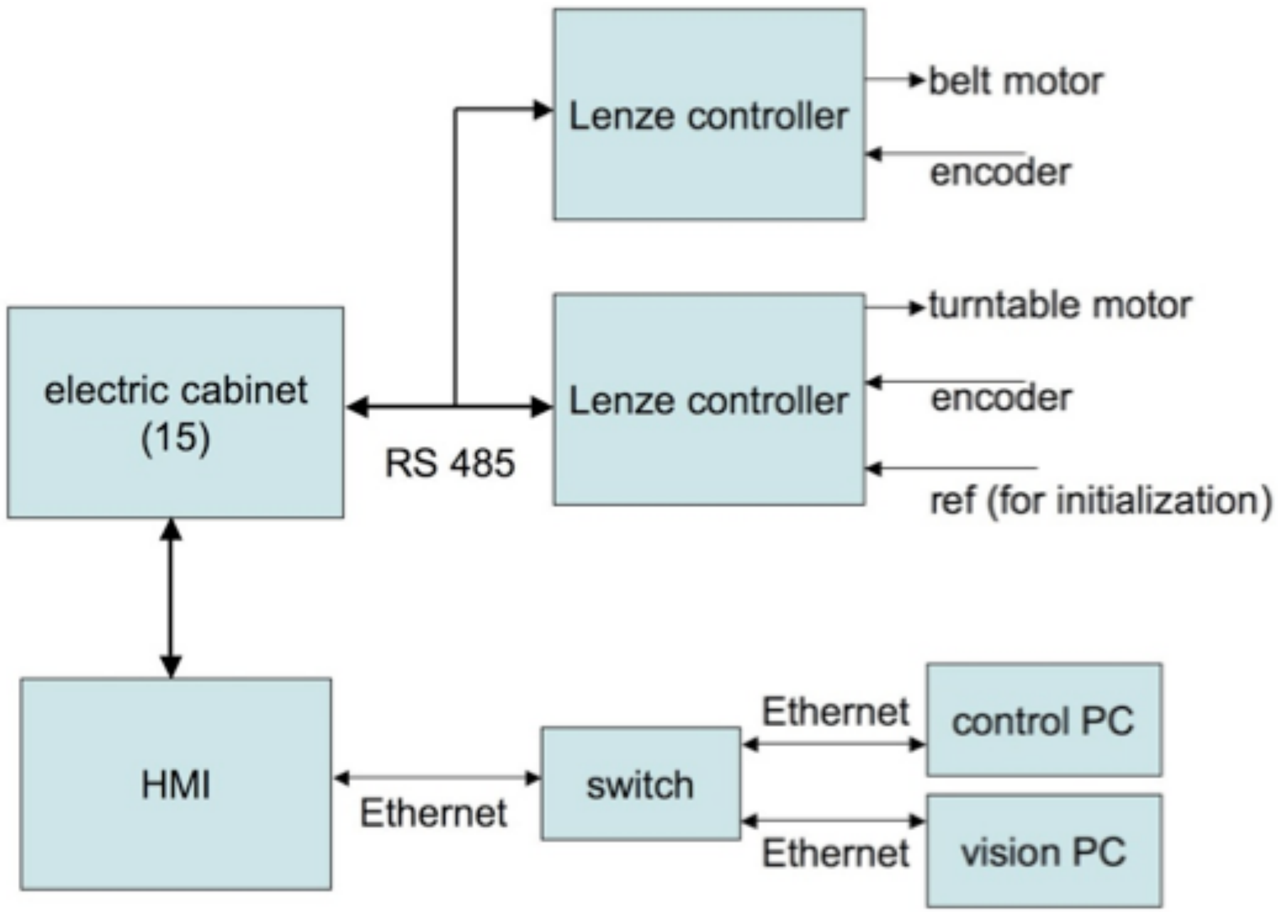


side view with mobile robot

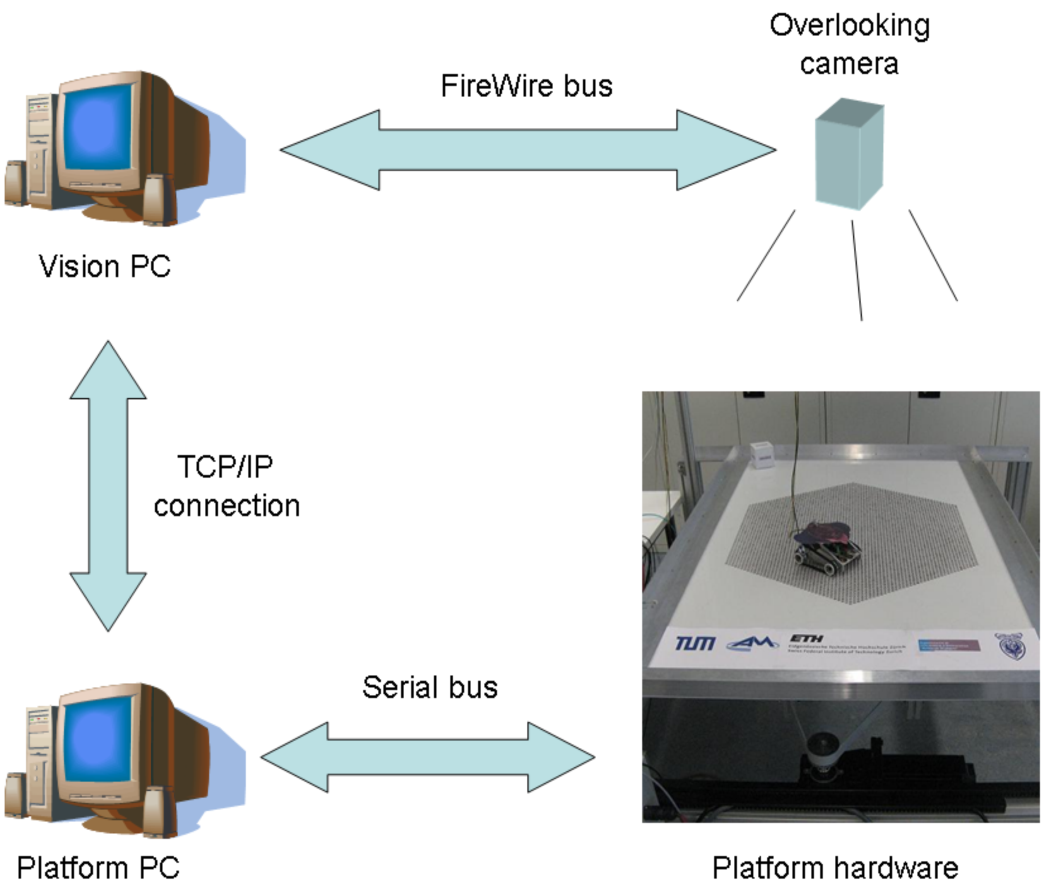


top view from the camera

Scheme of control hardware

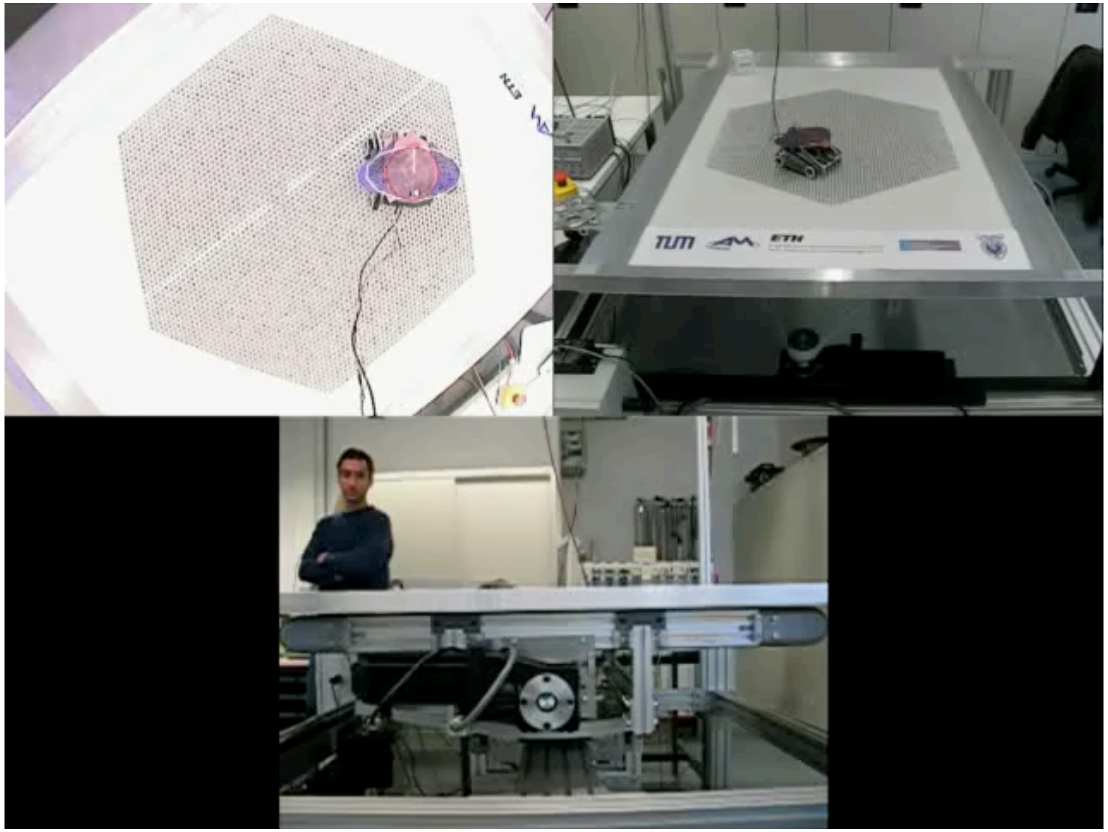


Experiments
ball-array platform



- position extraction data rate: 10 Hz
- velocity commands data rate: 10 Hz
- walking user replaced by a remote-controlled car
- different scenarios
  - standing still, but initially out of center
  - moving at constant velocity
  - traveling on a circular path
  - traveling on a square path

Standing still



video

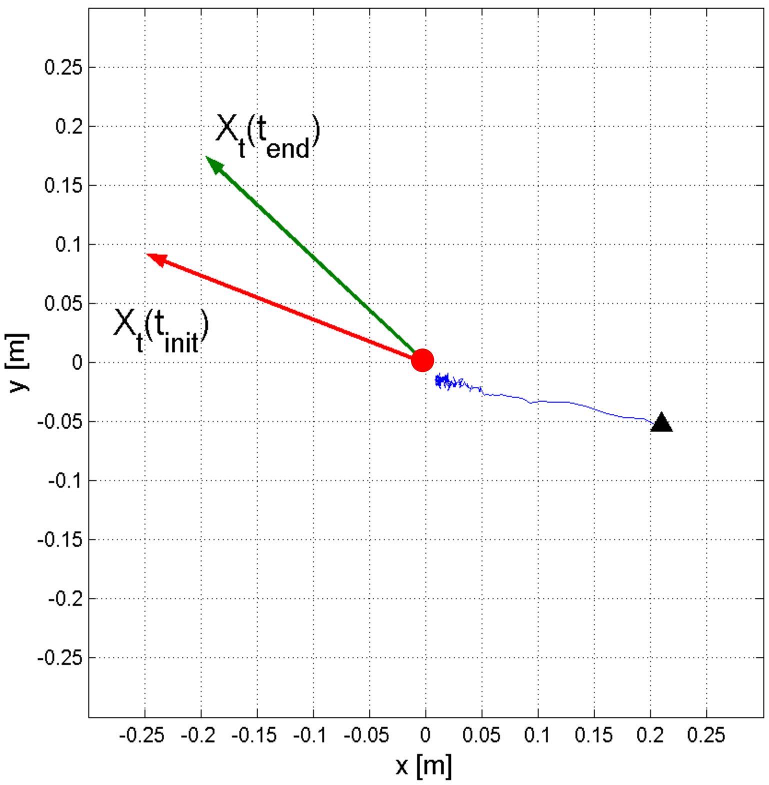
- no need of velocity compensation
- control parameters (used everywhere)

$$k = 4$$

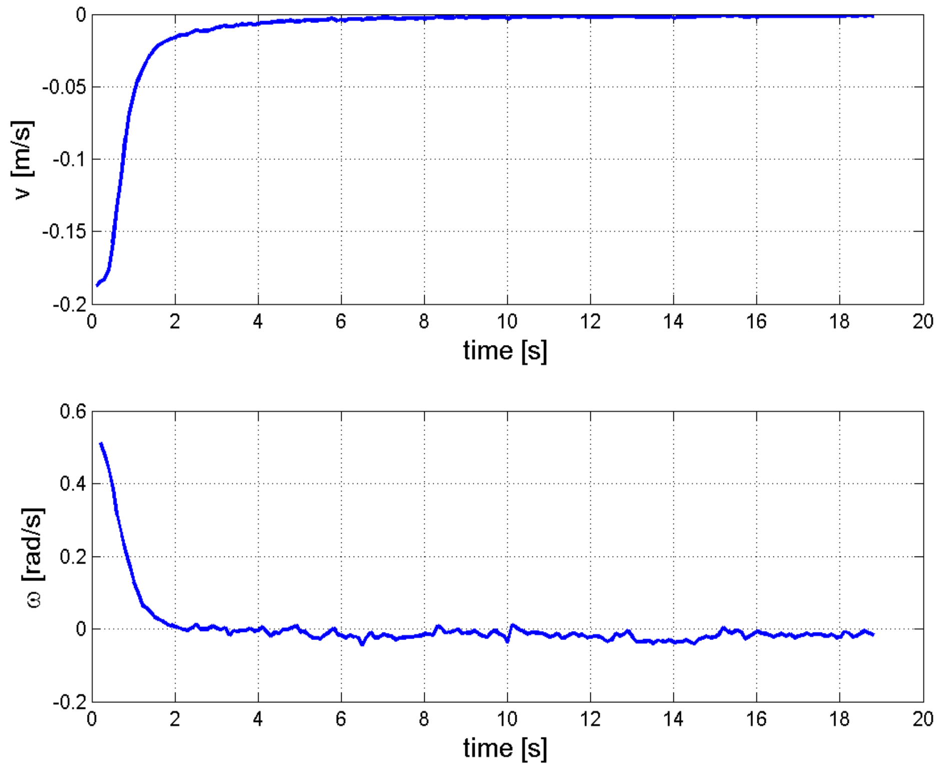
$$k_w = 0.3$$

$$\omega_{ff,max} = 0.05 \text{ rad/s}$$

Standing still

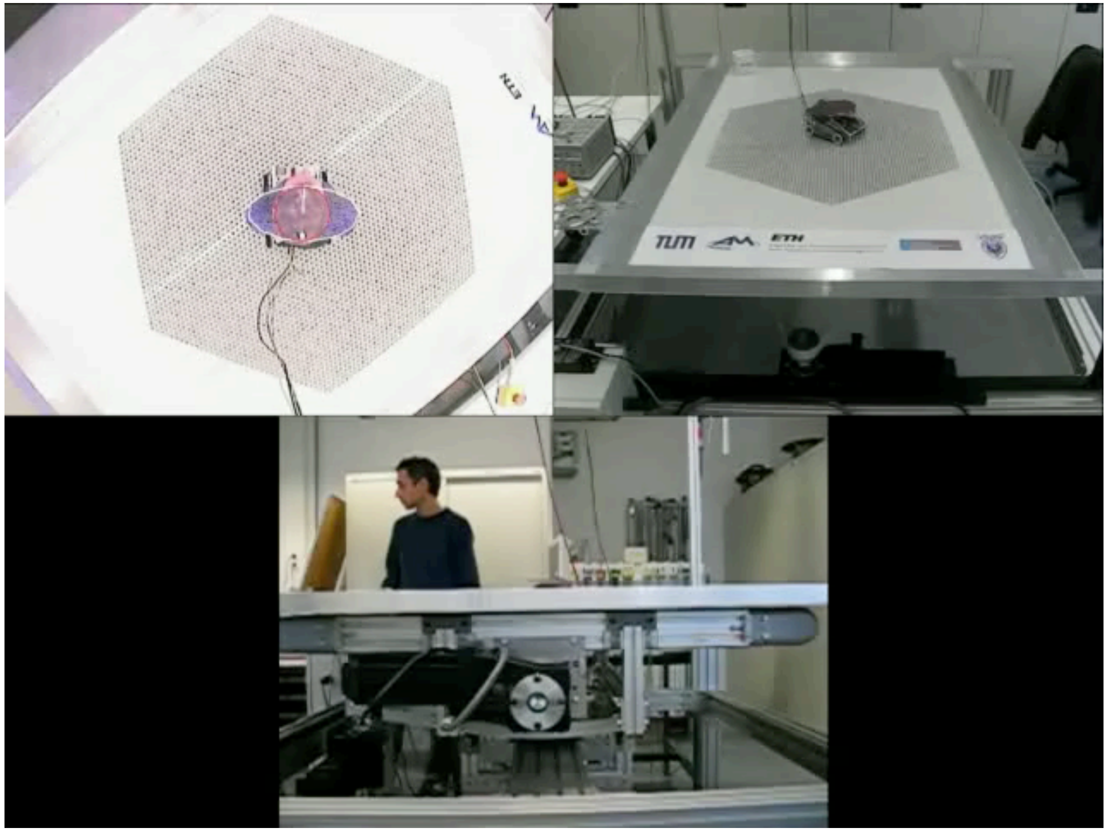


absolute trajectory



platform velocity commands

Moving at constant velocity

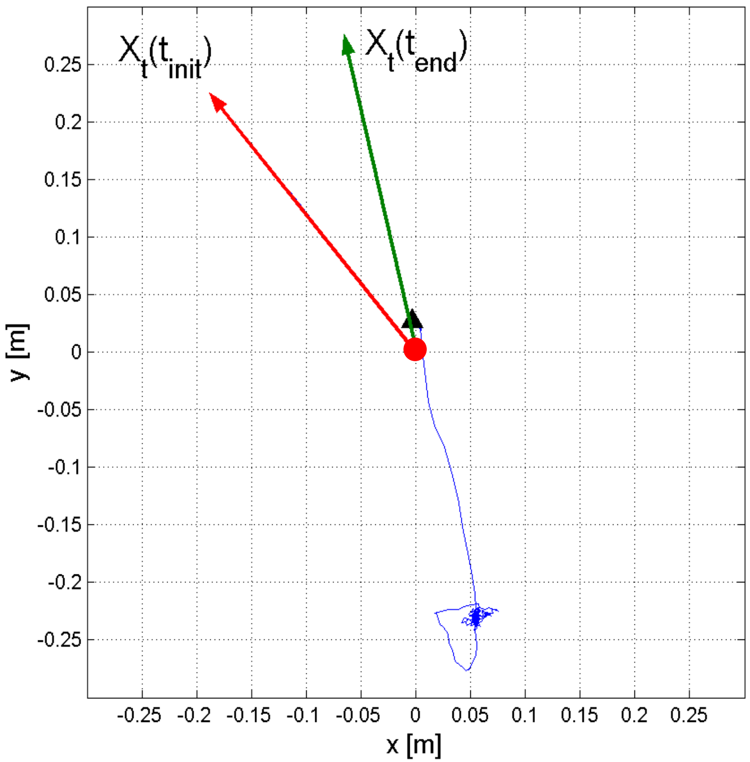


video

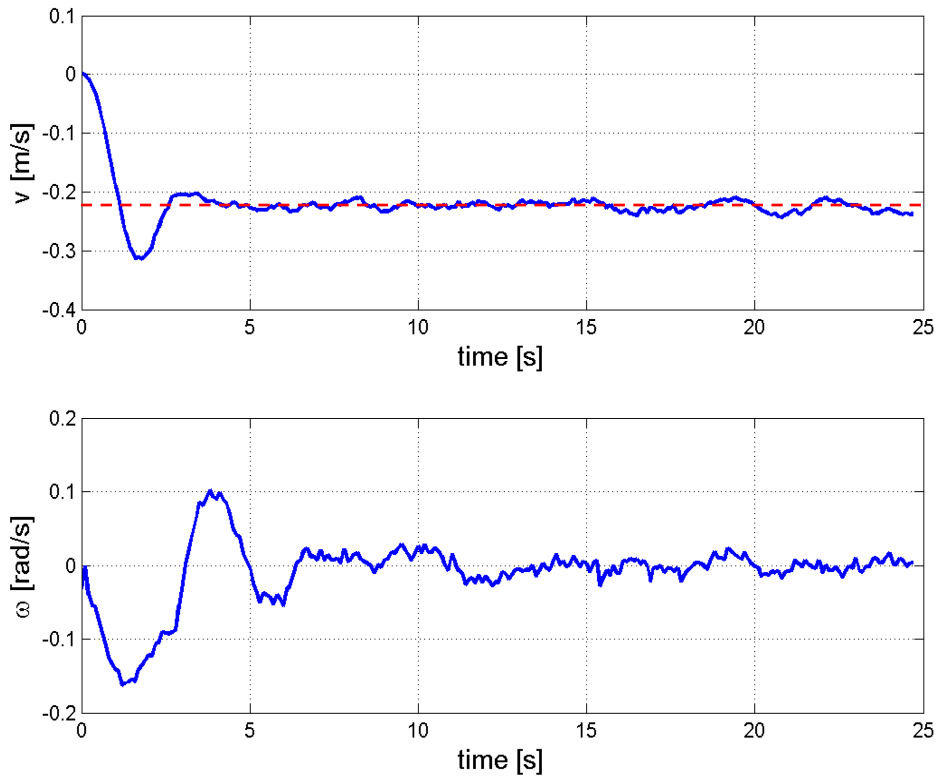
- user velocity ~ 0.22 m/s
- without compensation of intentional velocity

Moving at constant velocity

- without compensation of intentional velocity

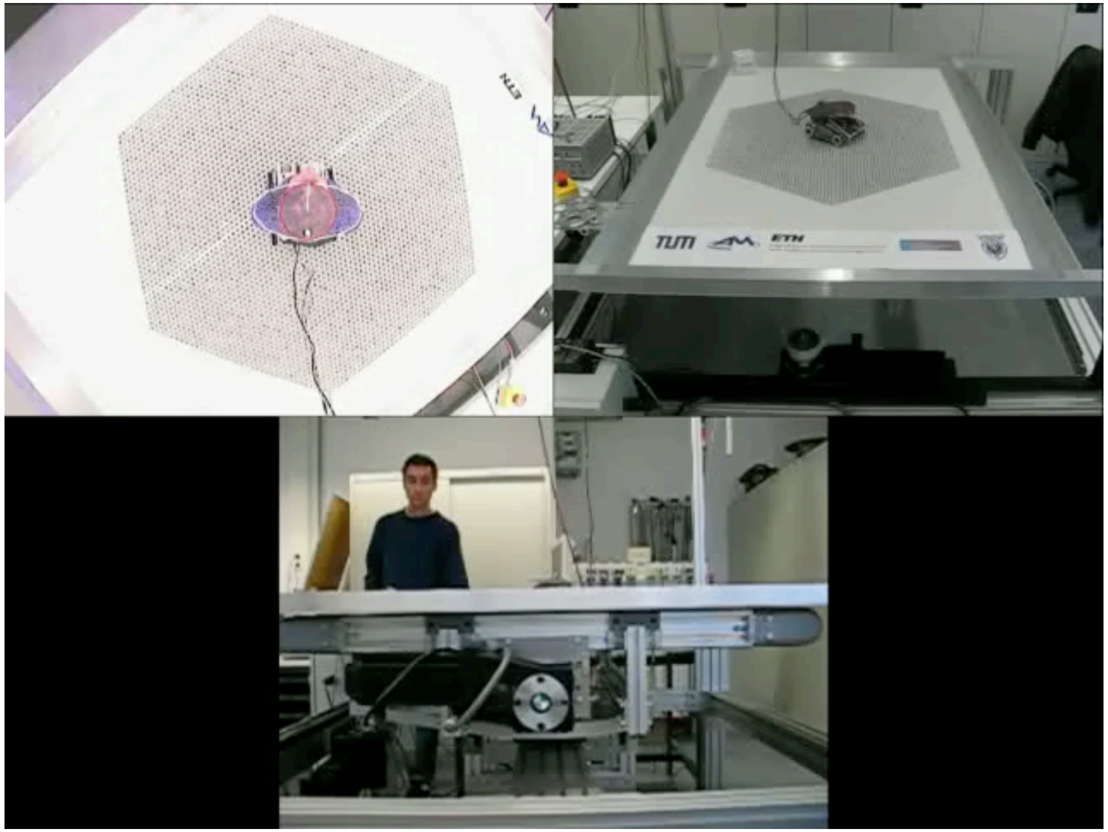


absolute trajectory



platform velocity commands

Moving at constant velocity

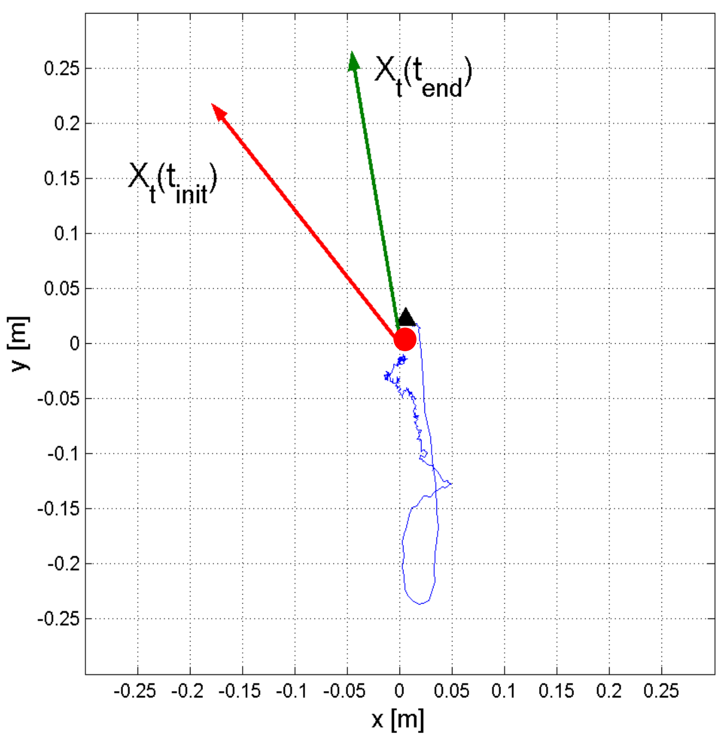


video

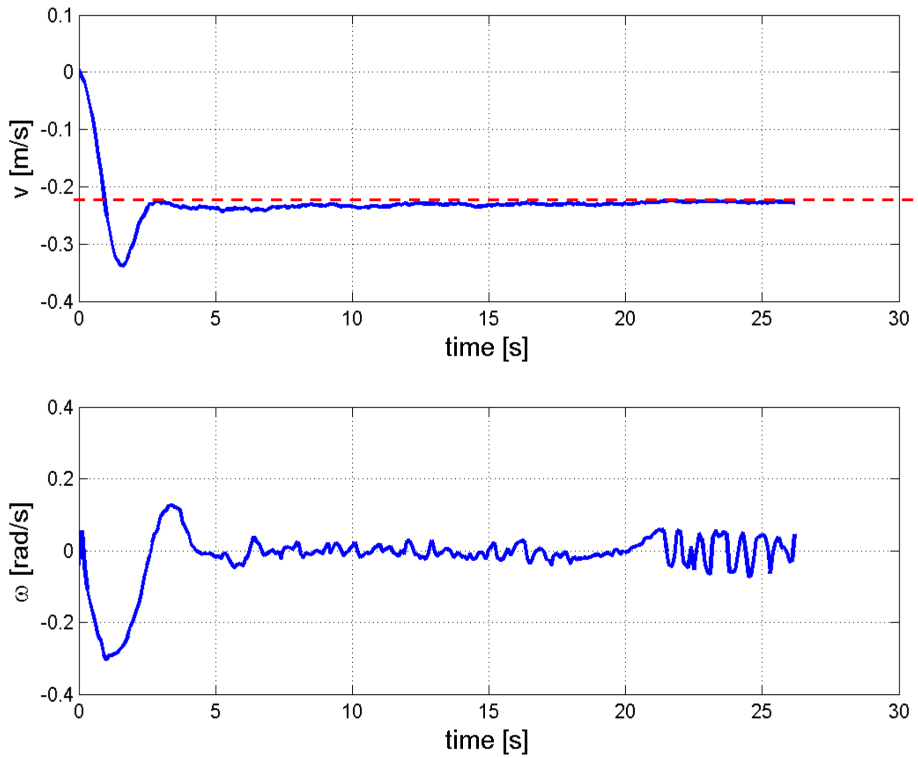
- user velocity ~ 0.22 m/s
- <u>with</u> feedforward compensation of intentional velocity

Moving at constant velocity

- <u>with</u> feedforward compensation of intentional velocity



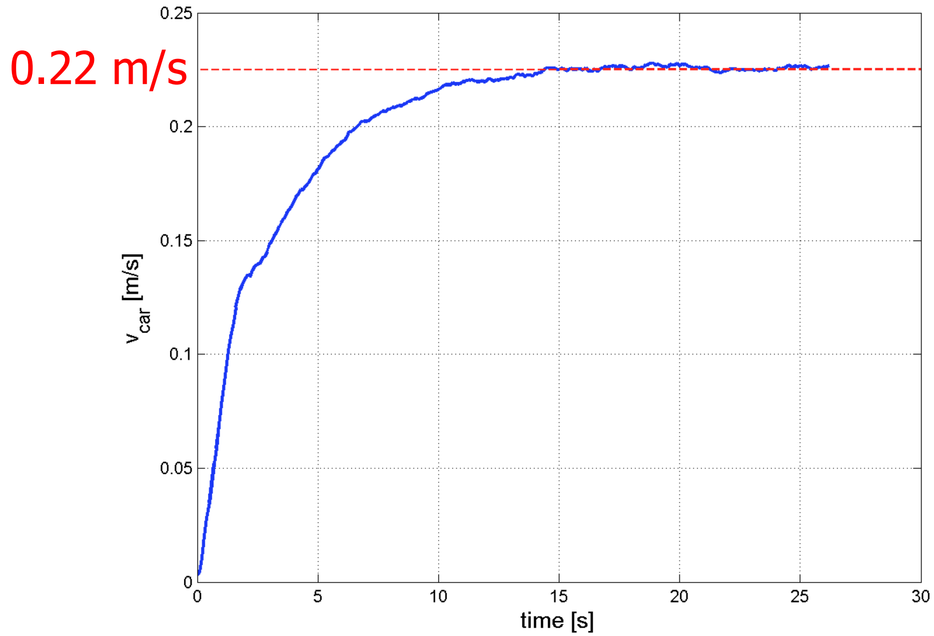
absolute trajectory



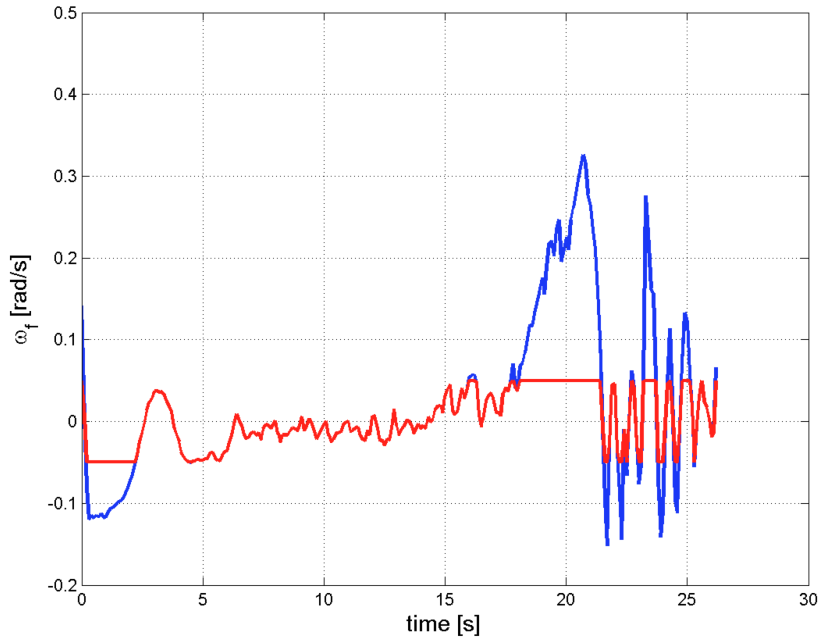
platform velocity commands

Moving at constant velocity

- estimation of the intentional speed



- angular feedforward term with and without saturation



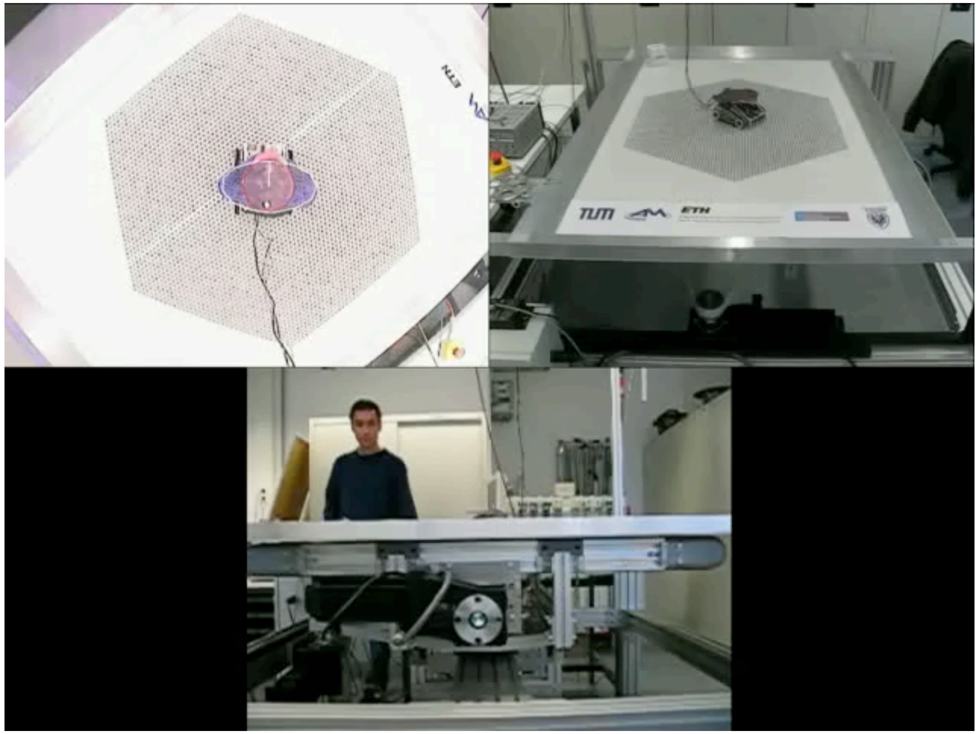
$$\omega_{ff,max} = 0.05 \text{ rad/s}$$

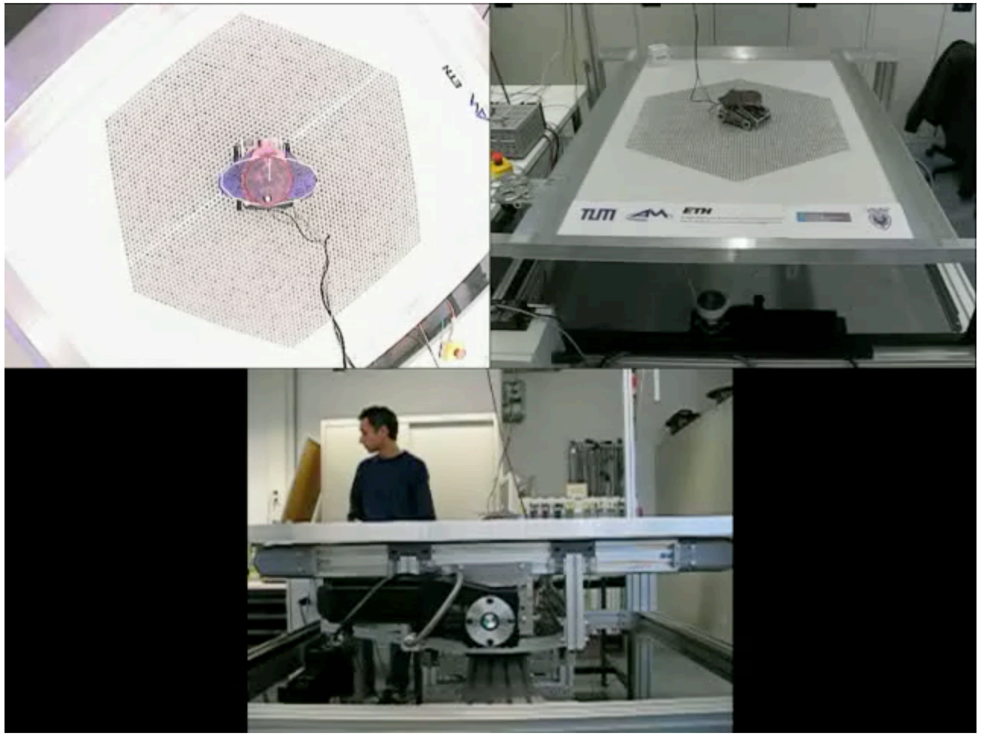
Moving on circular path

of radius = 0.35 m, at speed ≈ 0.14 m/s, stop after about 23.5 s

video

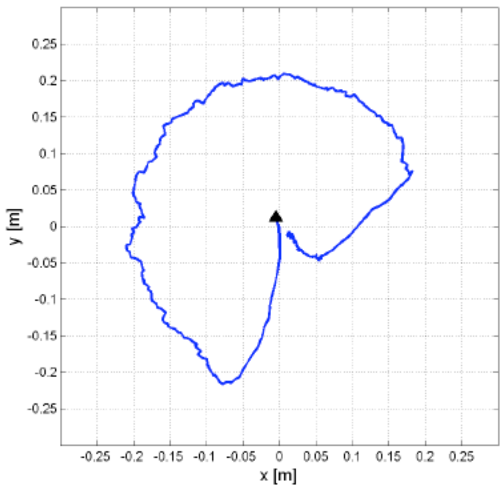
video



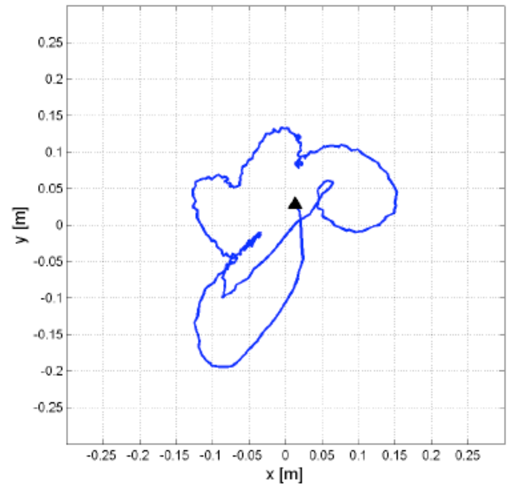


- pure feedback without compensation
- with feedforward compensation of intentional velocity

Moving on circular path



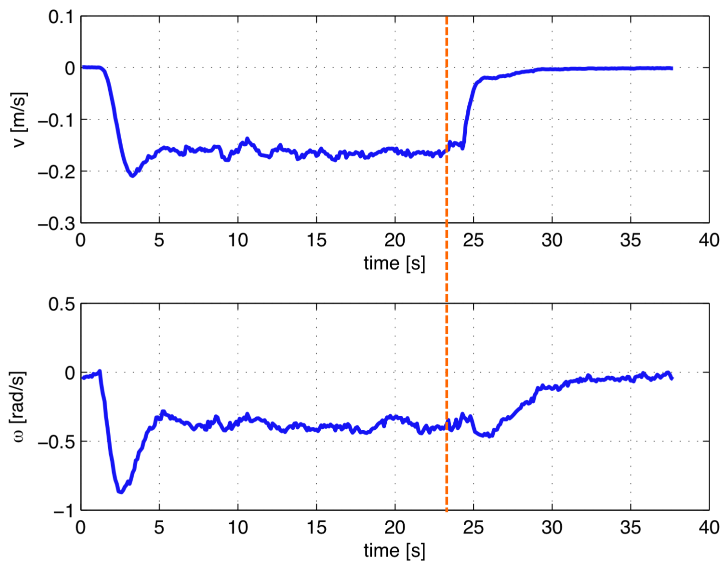
pure feedback

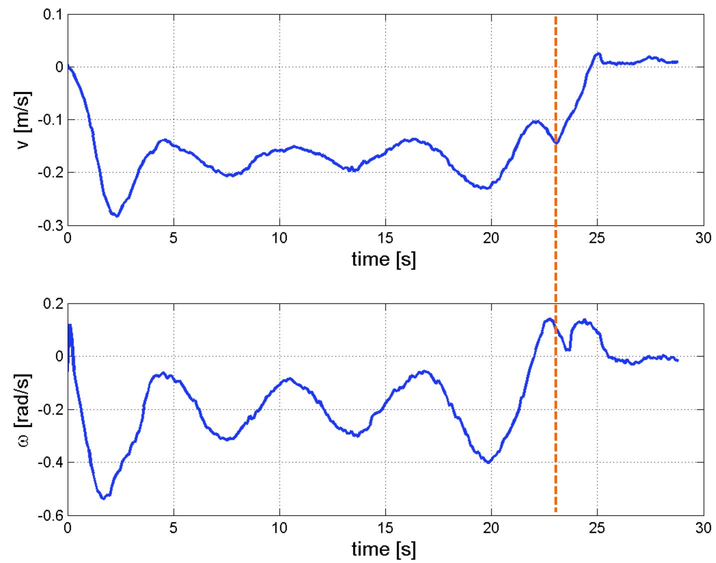


feedback + feedforward

linear velocity

angular velocity



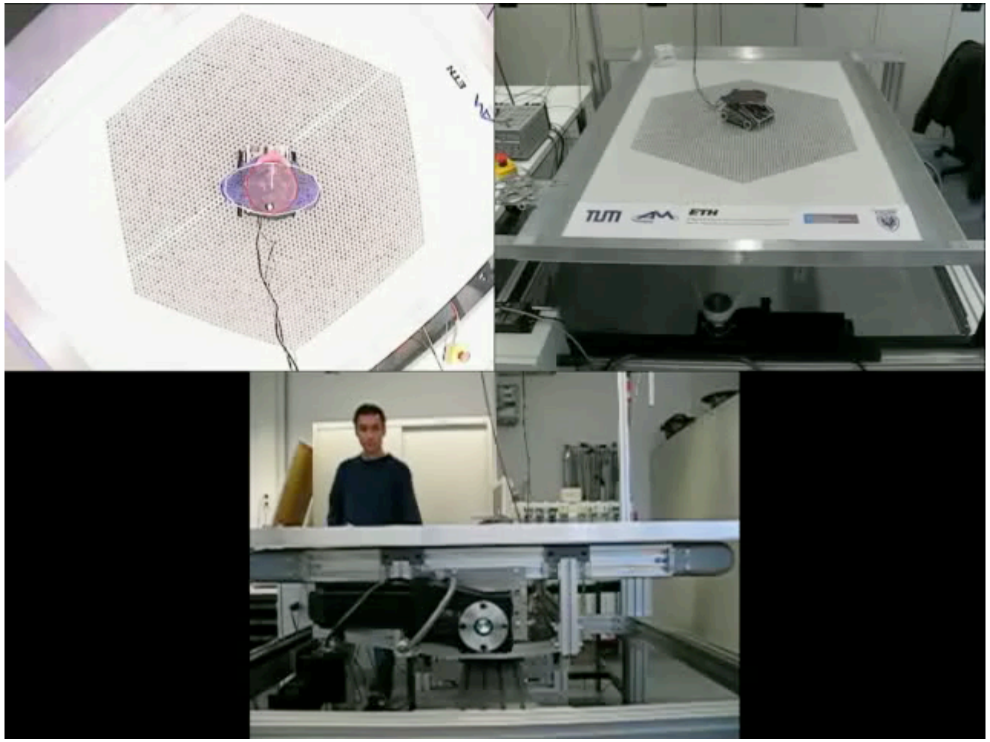


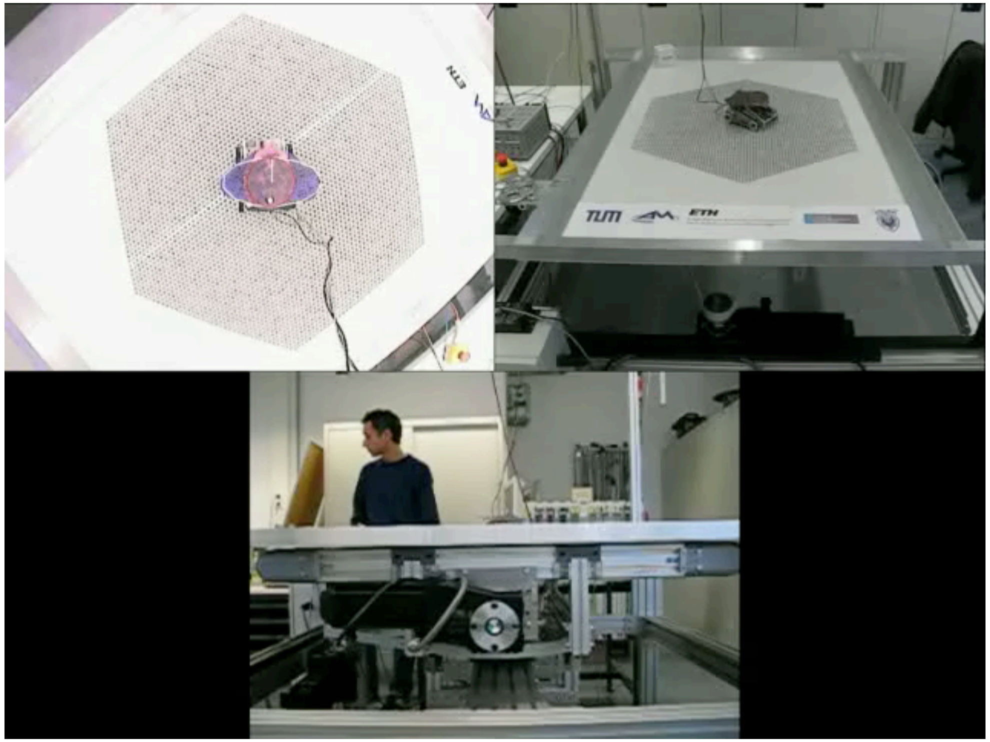
Moving on square path

of side = 0.4 m, at speed ≈ 0.1 m/s, turning at corners with $\omega = \pi/4$ rad/s

video

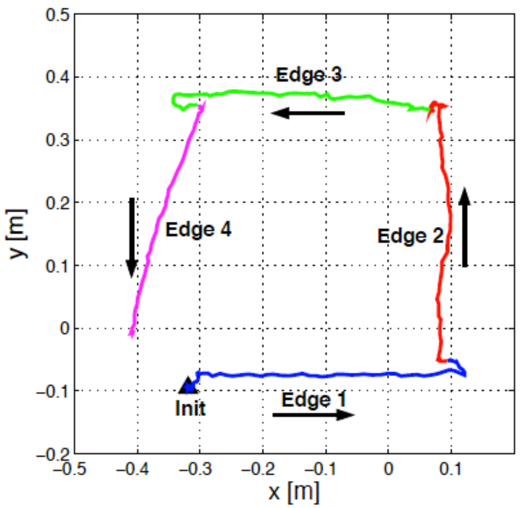
video



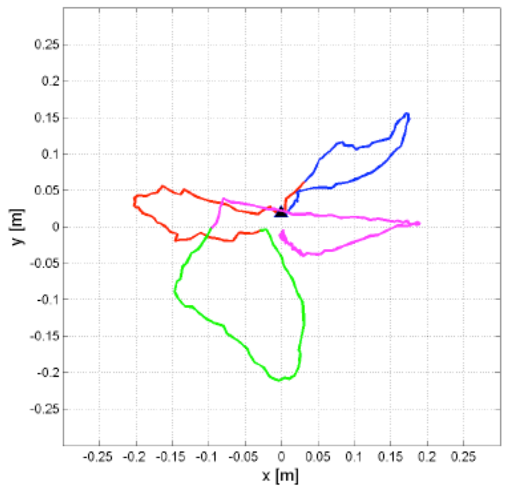


- pure feedback without compensation
- with feedforward compensation of intentional velocity

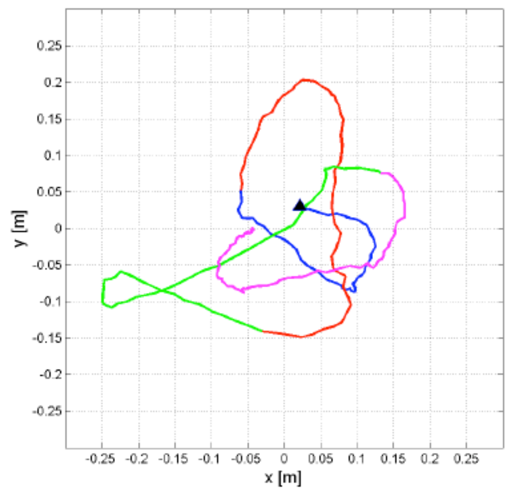
Moving on square path



(intentional) path



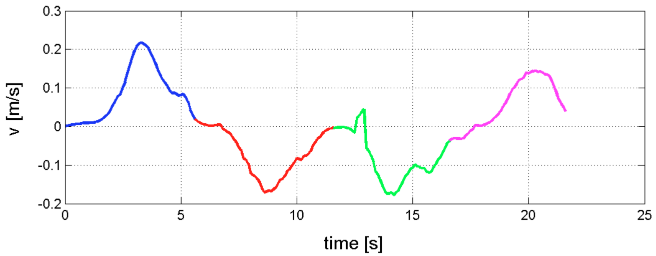
pure feedback

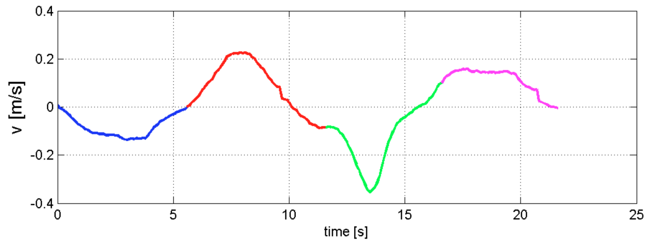


feedback + feedforward

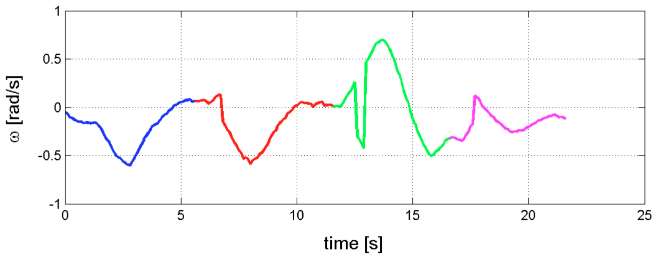
total motion $T = 24$ s

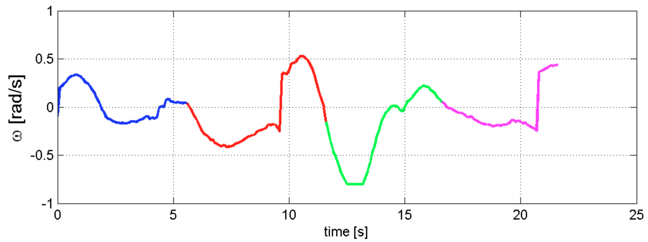
linear velocity





angular velocity





Acceleration-level control
ball-array platform

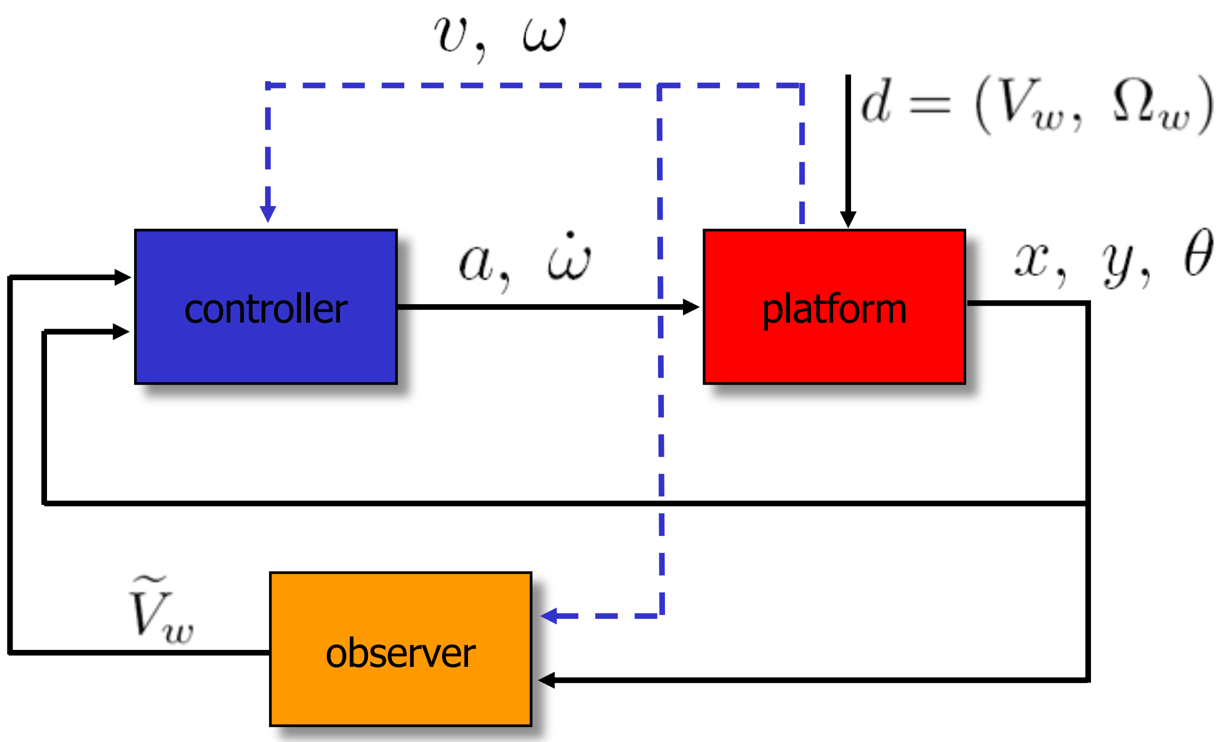
- why?
  - compliance with actuator limitations and perceptual comfort of the user (especially for full-scale case)
    - direct "control" over imposed accelerations
  - softer transients (no jumps in velocity)
  - allows analysis of dynamic effects on walker
- how?
  - add one integrator on each input in the model
  - extension of a first-order (velocity) smooth control law

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = f(x,\, y,\, \theta,\, \widetilde{V}_w)$$

Acceleration control design
ball-array platform

- a cascaded second-order control law

$$\begin{bmatrix} a \\ \dot{\omega} \end{bmatrix} = \frac{\mathrm{d}f(x,\, y,\, \theta,\, \widetilde{V}_w)}{\mathrm{d}t} - k_a\left(\begin{bmatrix} v \\ \omega \end{bmatrix} - f(x,\, y,\, \theta,\, \widetilde{V}_w)\right)$$



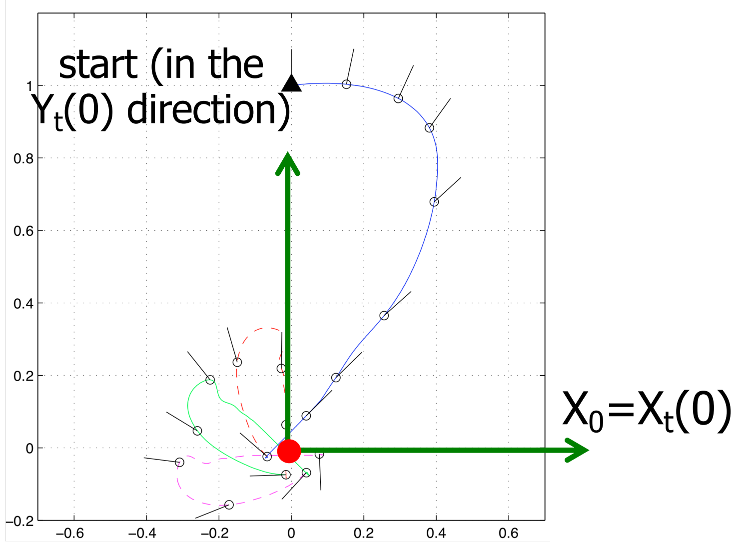
Simulation results
ball-array platform

- CyberCarpet under acceleration control
- walker: square path of side = 3 m, max velocity = 1.2 m/s (b-c-b acceleration)

video

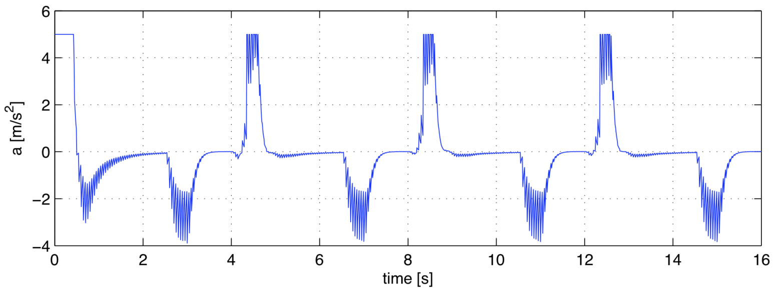
Simulation results
ball-array platform

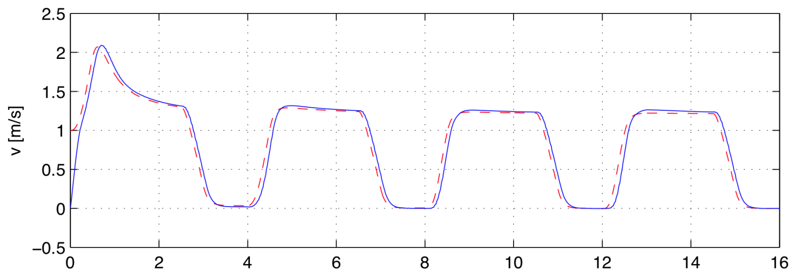
absolute trajectory under acceleration control



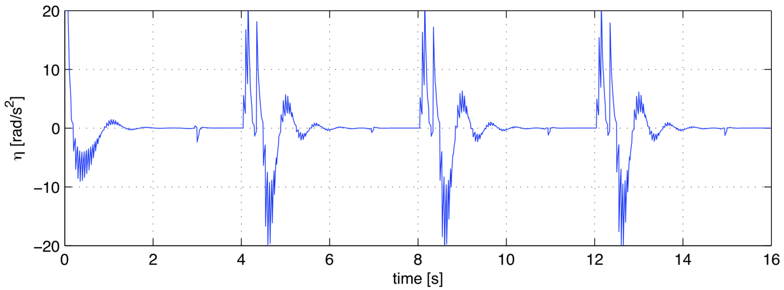
comparison between velocity control (---) and velocities resulting from acceleration control (i.e., after integration) with k_a = diag{20,20}

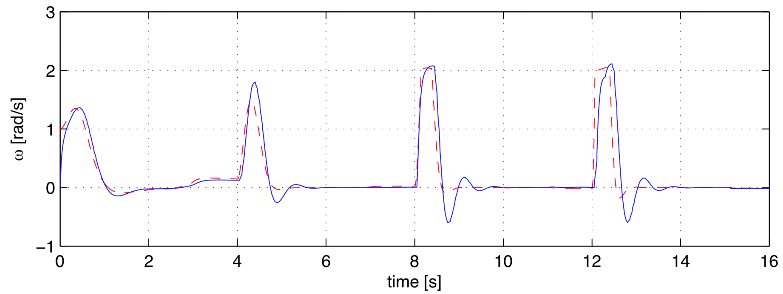
linear acceleration





angular acceleration

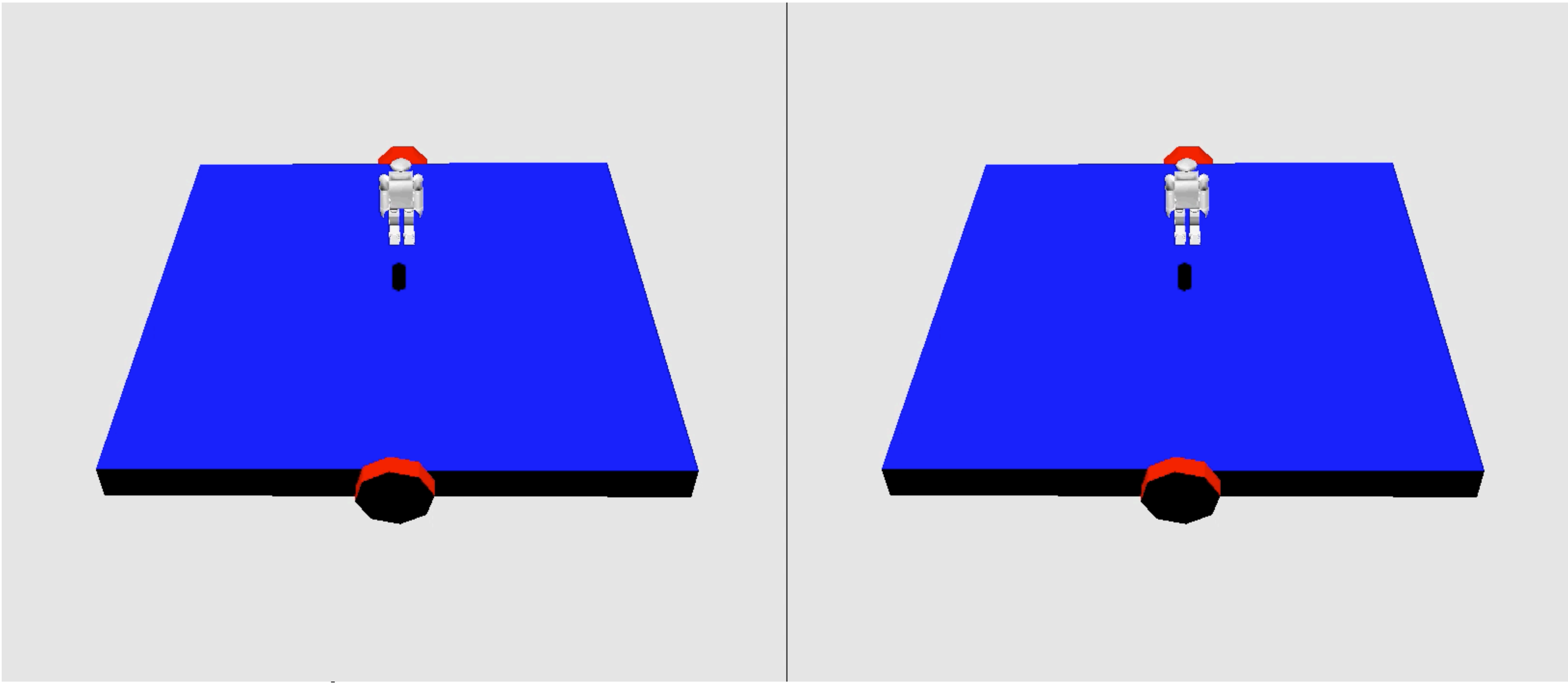




Dynamic analysis
ball-array platform

- CyberCarpet under acceleration control
- walker: square path of side = 3 m, max velocity = 1.2 m/s (b-c-b acceleration)

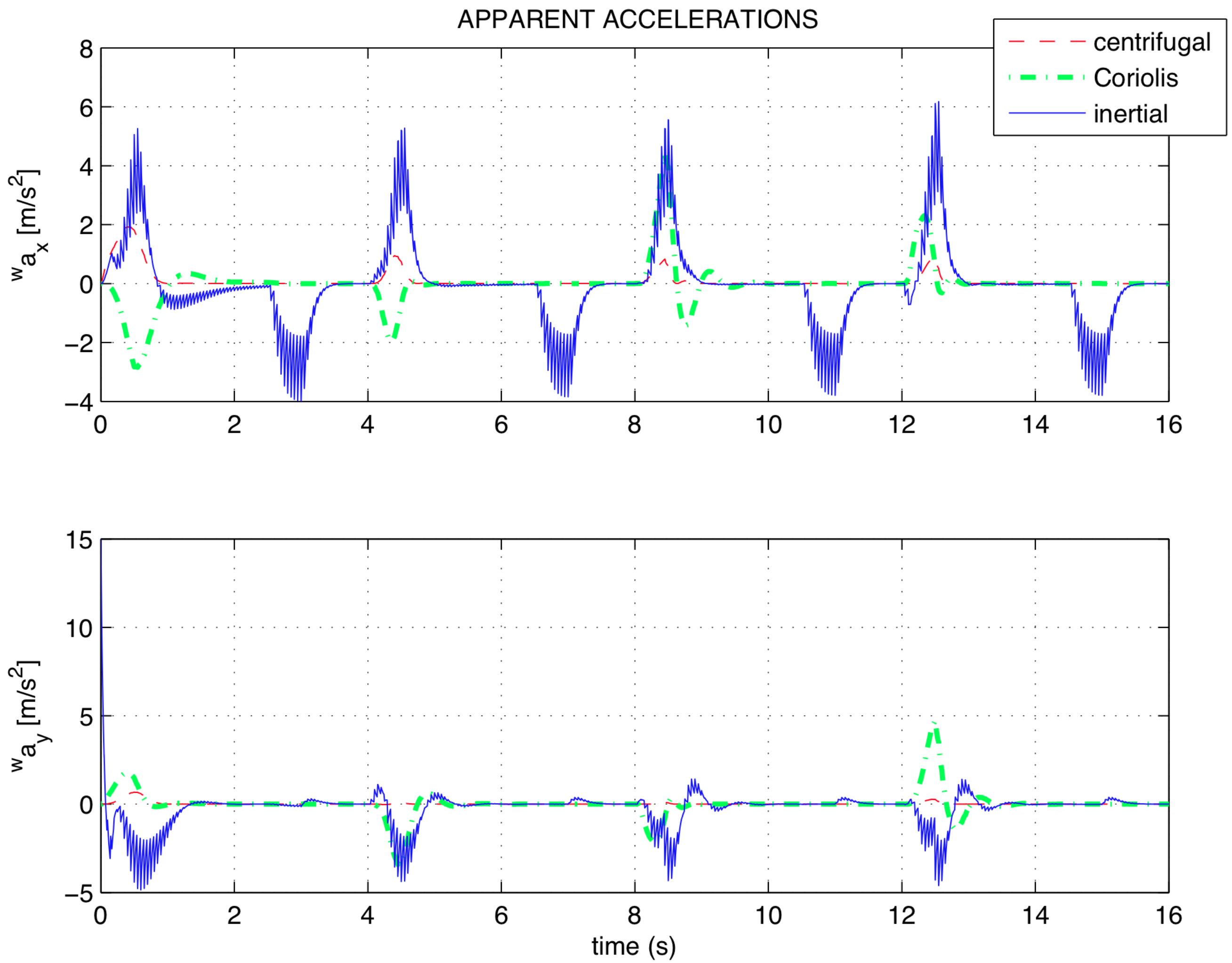
video



inertial acceleration   Coriolis acceleration   centrifugal acceleration

Apparent accelerations
ball-array platform

acting on
the walker



Bibliography

- A. De Luca, R. Mattone, and P. Robuffo Giordano, "The motion control problem for the CyberCarpet," *2006 IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, pp. 3532-3537, Orlando, 2006
- A. De Luca, R. Mattone, and P. Robuffo Giordano, "Feedback/feedforward schemes for motion control of the CyberCarpet," *8th IFAC Symp. on Robot Control (SYROCO'06)*, Bologna, 2006
- A. De Luca, R. Mattone, and P. Robuffo Giordano, "Acceleration-level control of the CyberCarpet," *2007 IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, pp. 2330-2335, Roma, 2007
- A. De Luca, R. Mattone, P. Robuffo Giordano, H. Ulbrich, M. Schwaiger, M. Van den Bergh, E. Koller-Meier, and L. Van Gool, "Motion control of the CyberCarpet platform," *IEEE Trans. on Control Systems Technology*, vol. 21, no. 2, pp. 410-427, 2013 [on-line in IEEE Xplore since 6 Feb 2012]

Updated web links
(as of March 2021)

working links about the CyberWalk project

- https://www.kyb.tuebingen.mpg.de/149540/perception-for-action (MPI, Biological Cybernetics)
- https://www.uni-ulm.de/in/psy-acog/forschung/projekte/former-3rd-party-projects/cyberwalk/ (Marc Ernst, project coordinator)
- https://www.mw.tum.de/en/am/research/completed-projects/cyberwalk (TUM, Applied Mechanics)
- http://www.diag.uniroma1.it/labrob/research/CW.html (Sapienza, DIAG)