Learning Periodic Human Motion through Imitation using Eigenposes

by

Rawichote Chalodhorn Ph.D.

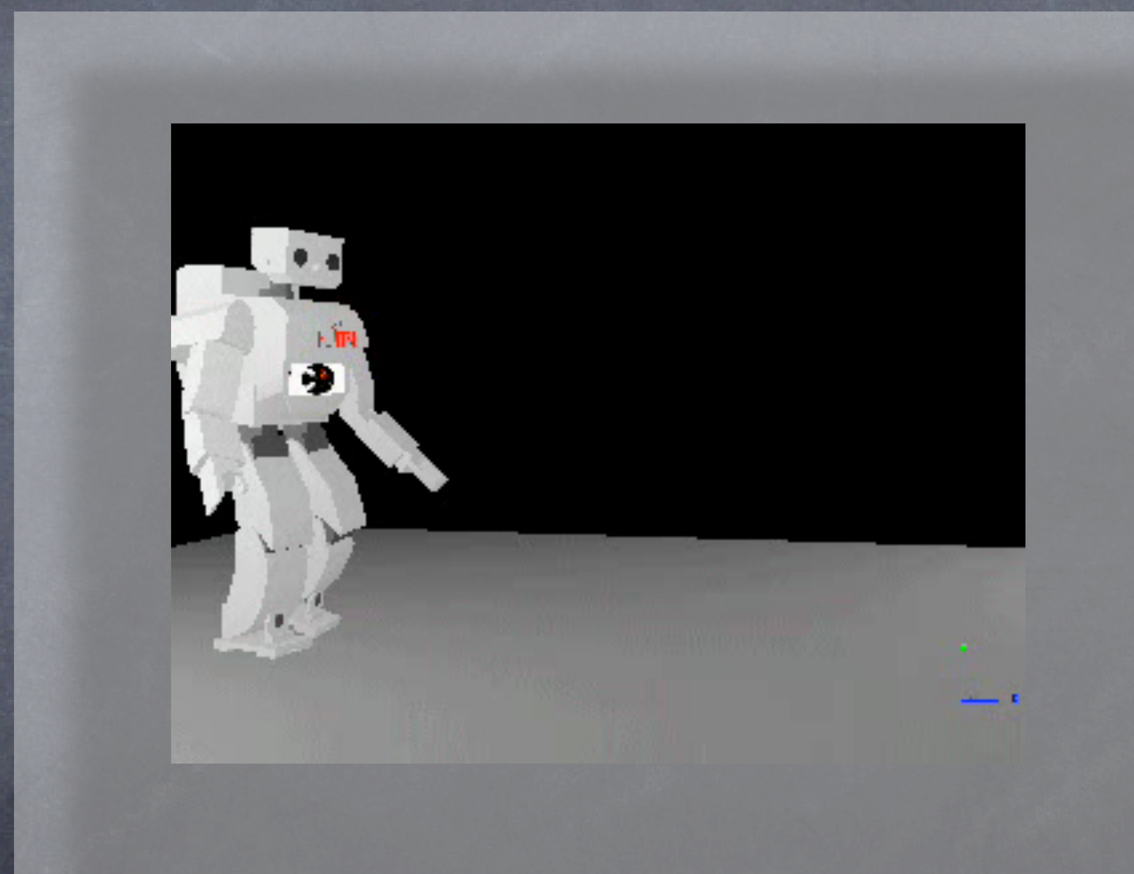
Humanoid Robotics Lab, Neural System Group,

Dept. of Computer Science & Engineering, University of Washington.

The ultimate goal is to learn a complex task by imitation

Q: Can we just replay the motion?

A: Apparently not!



Problem No.1

The motion pattern needs to be optimized to match the dynamics of the robot.

But! Direct optimization of full-body "high-dimensional" joint angle data is "intractable".

Problem No.2

We bought a commercial robot, but the company just simply doesn't give us the dynamic model.

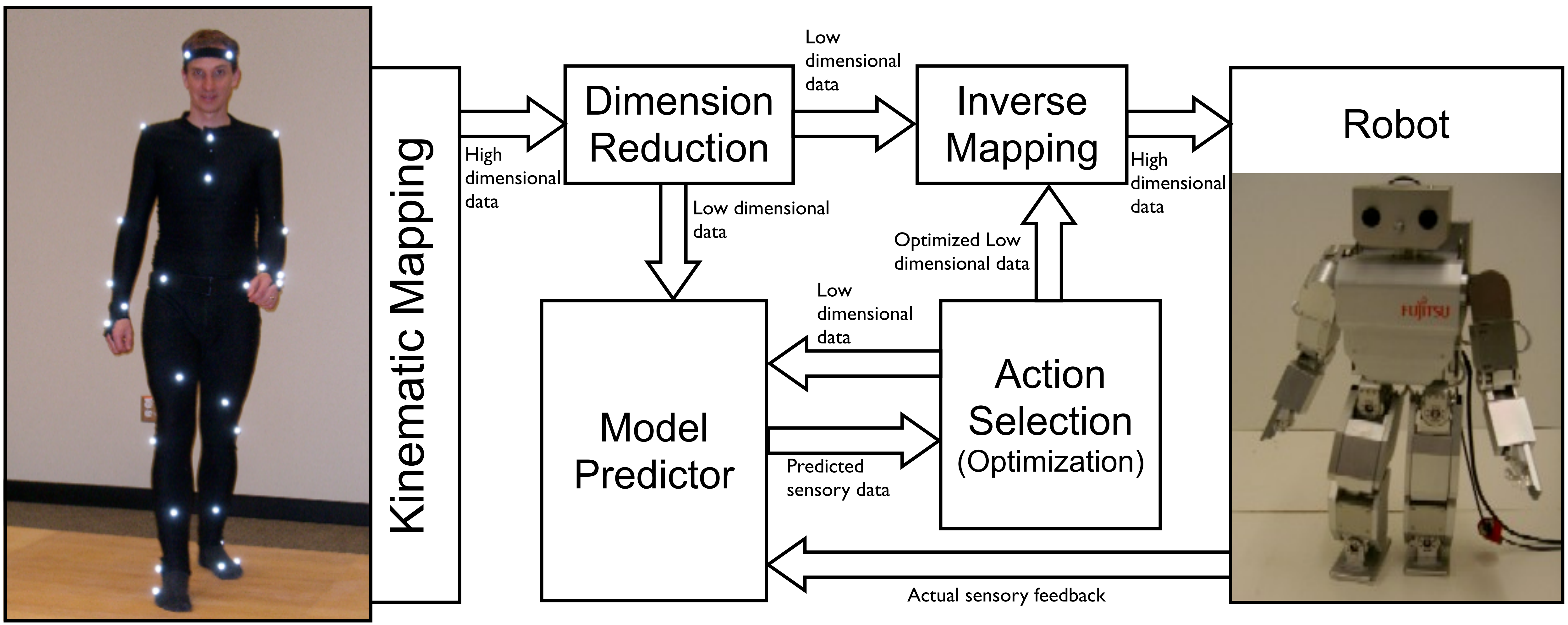
What should we do?

The dynamic model "is not" available!

Research statement

The research goal is to "generate full-body humanoid motions" while the problem of "intractable of high dimensional data" is inherited and the problem of "absences of dynamic model" is presence.

Proposed framework



Presentation outline

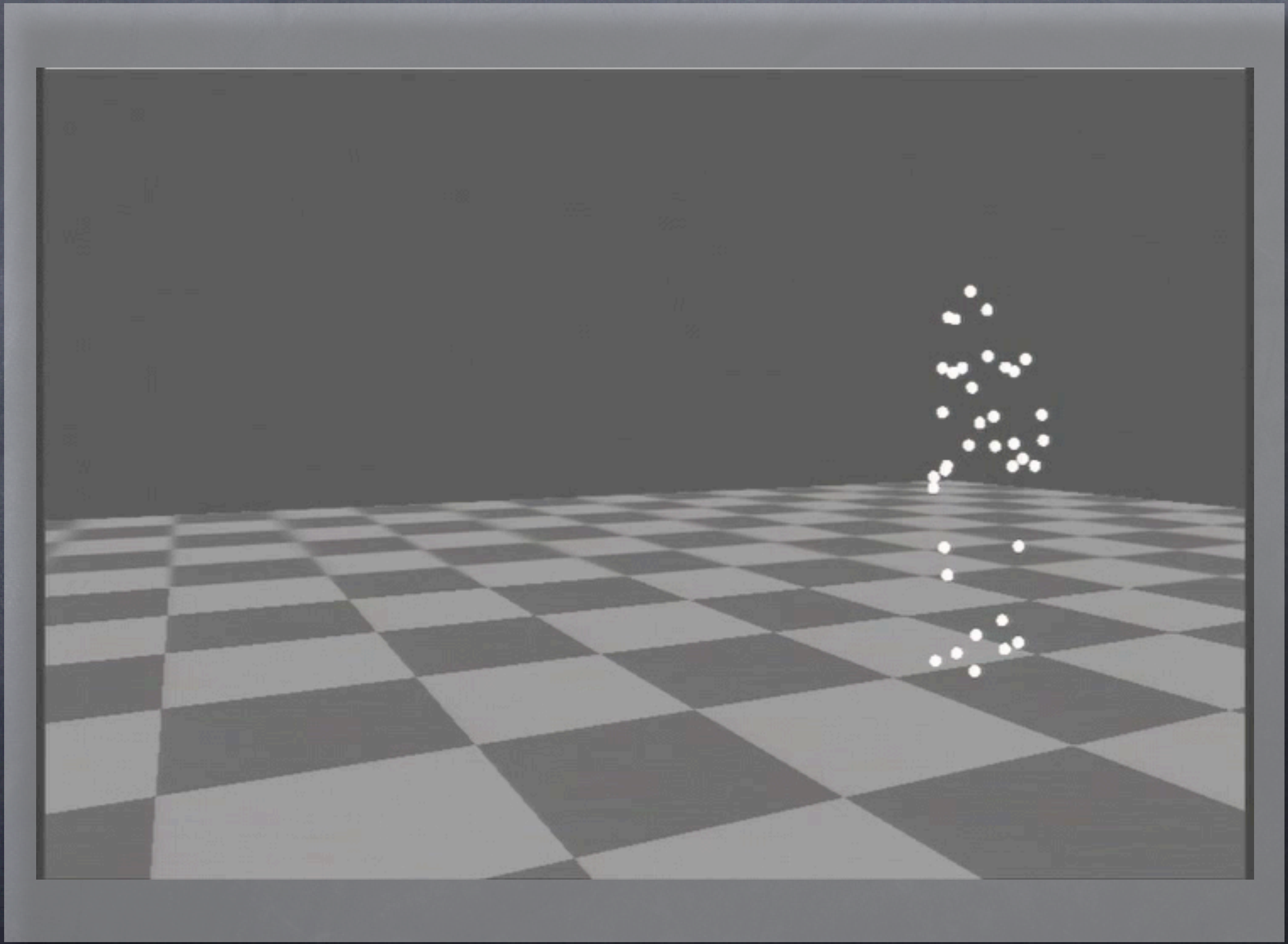
- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation

Dimension reduction algorithms

- Linear Principal components analysis (PCA)
  [Karhunen and Loève 1940s']
- None-Linear PCA [Kirby and Miranda, 1996]
- Locally Linear Embedding (LLE)
  [Roweis and Saul, 2000]
- ISOMAP [Tenenbaum et al., 2000]
- Gaussian Process Latent Variable Models
  [Neil D. Lawrence 2003]

Low Dimensional posture space

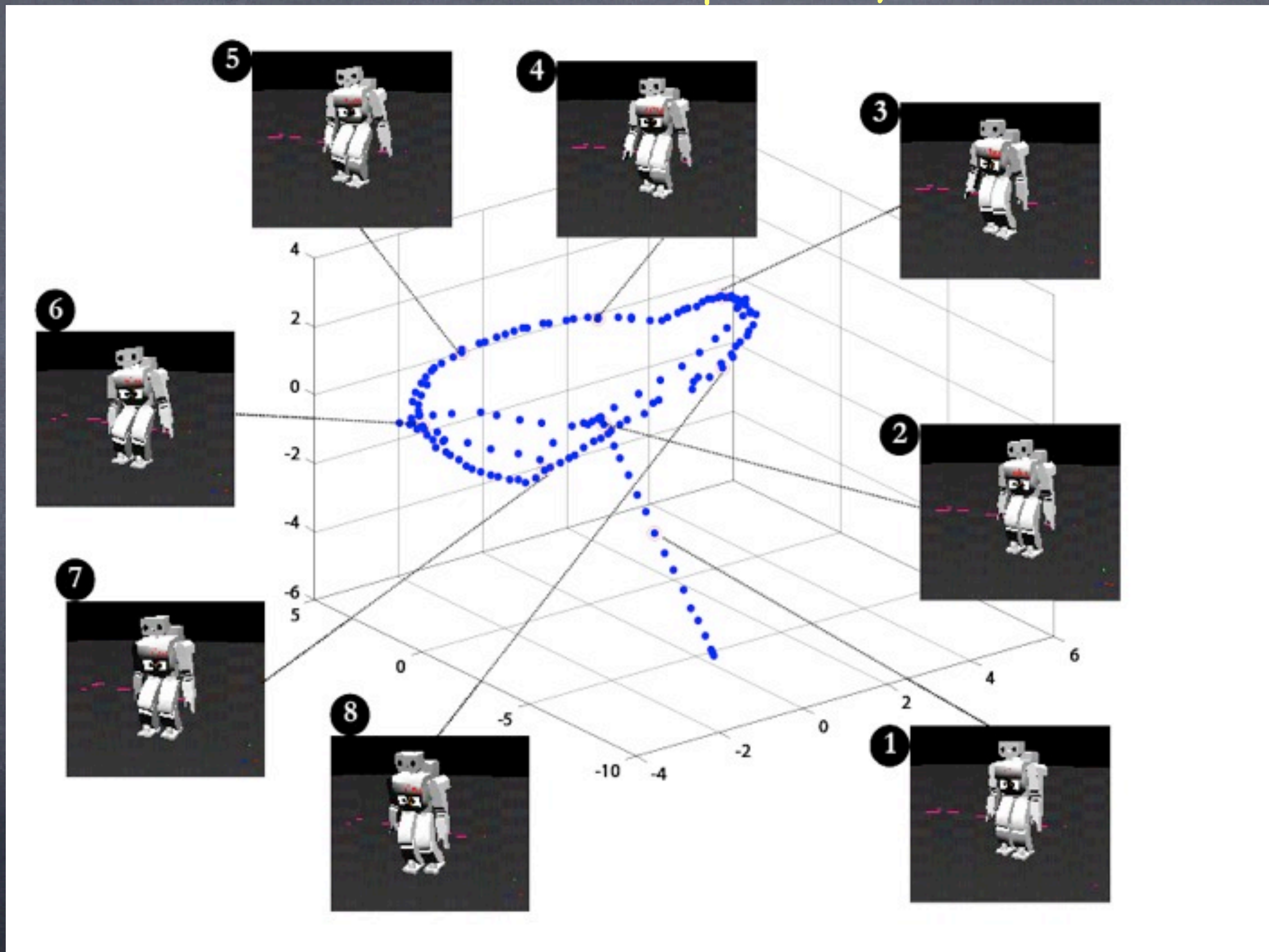
[Gaussian Process Latent Variable Models]



Courtesy of Keith Grochow

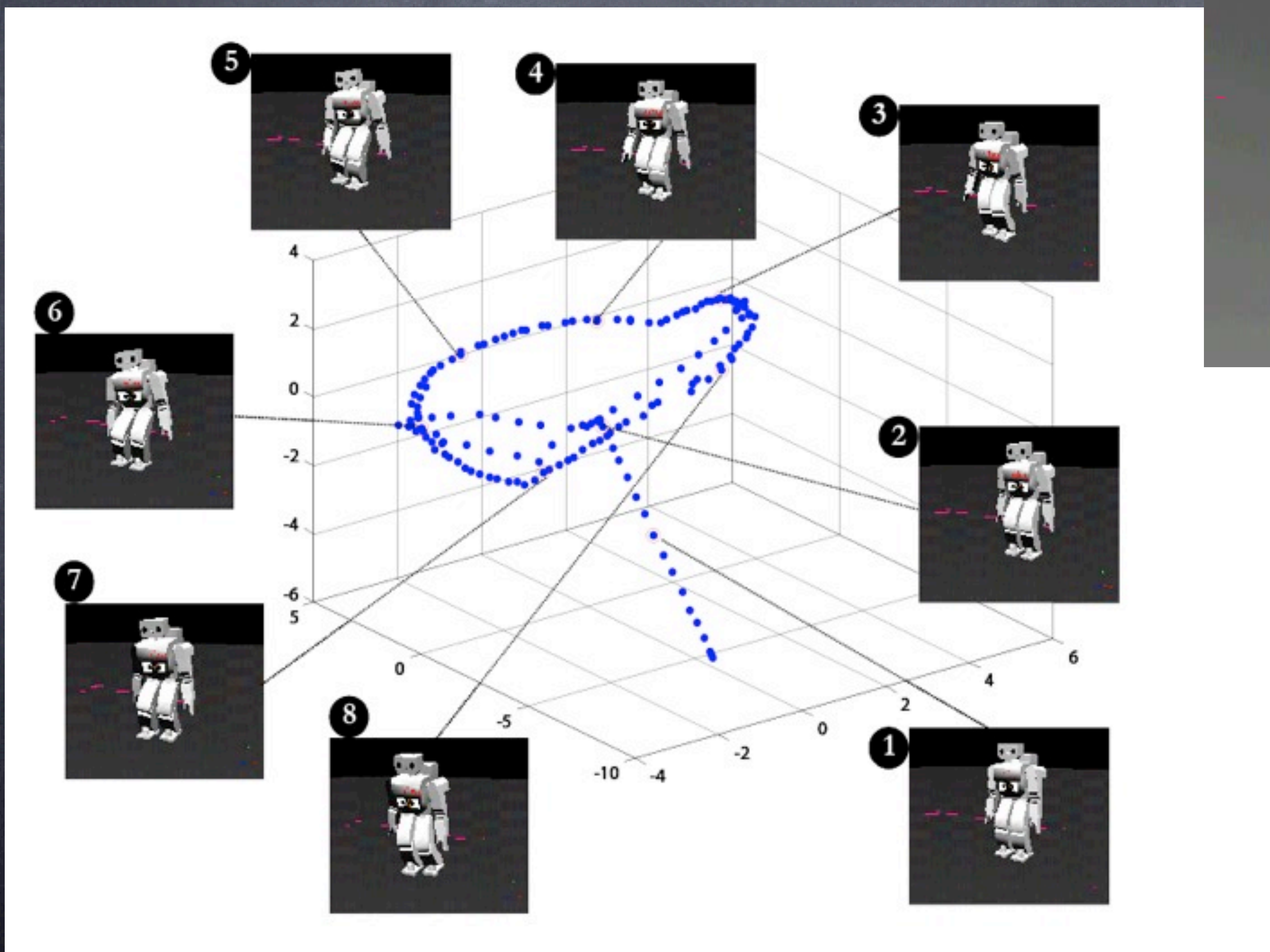
The “eigenpose” space

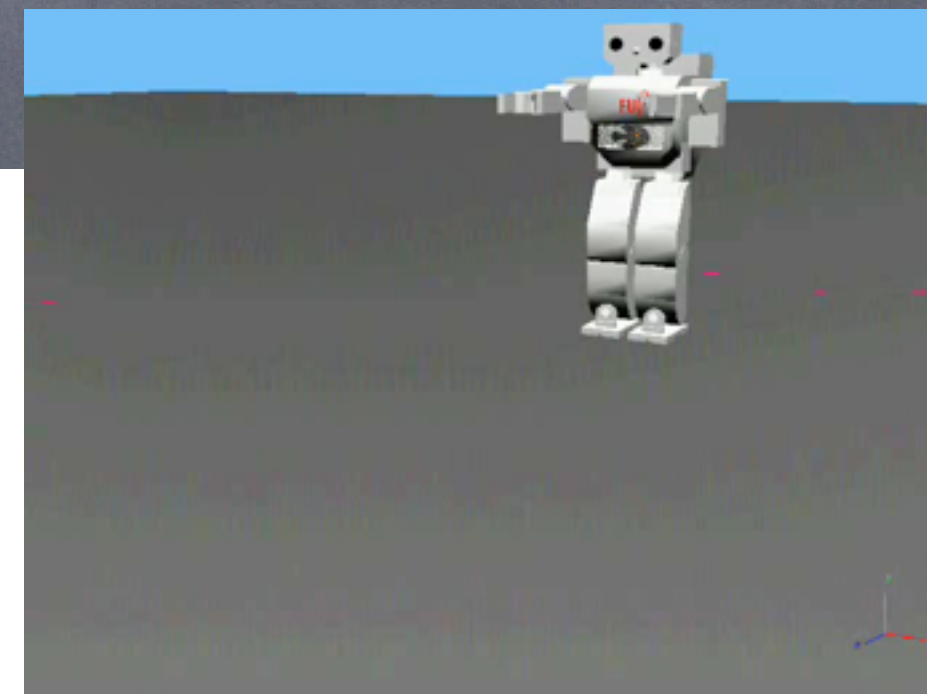
3-D low-dimensional subspaces by linear PCA



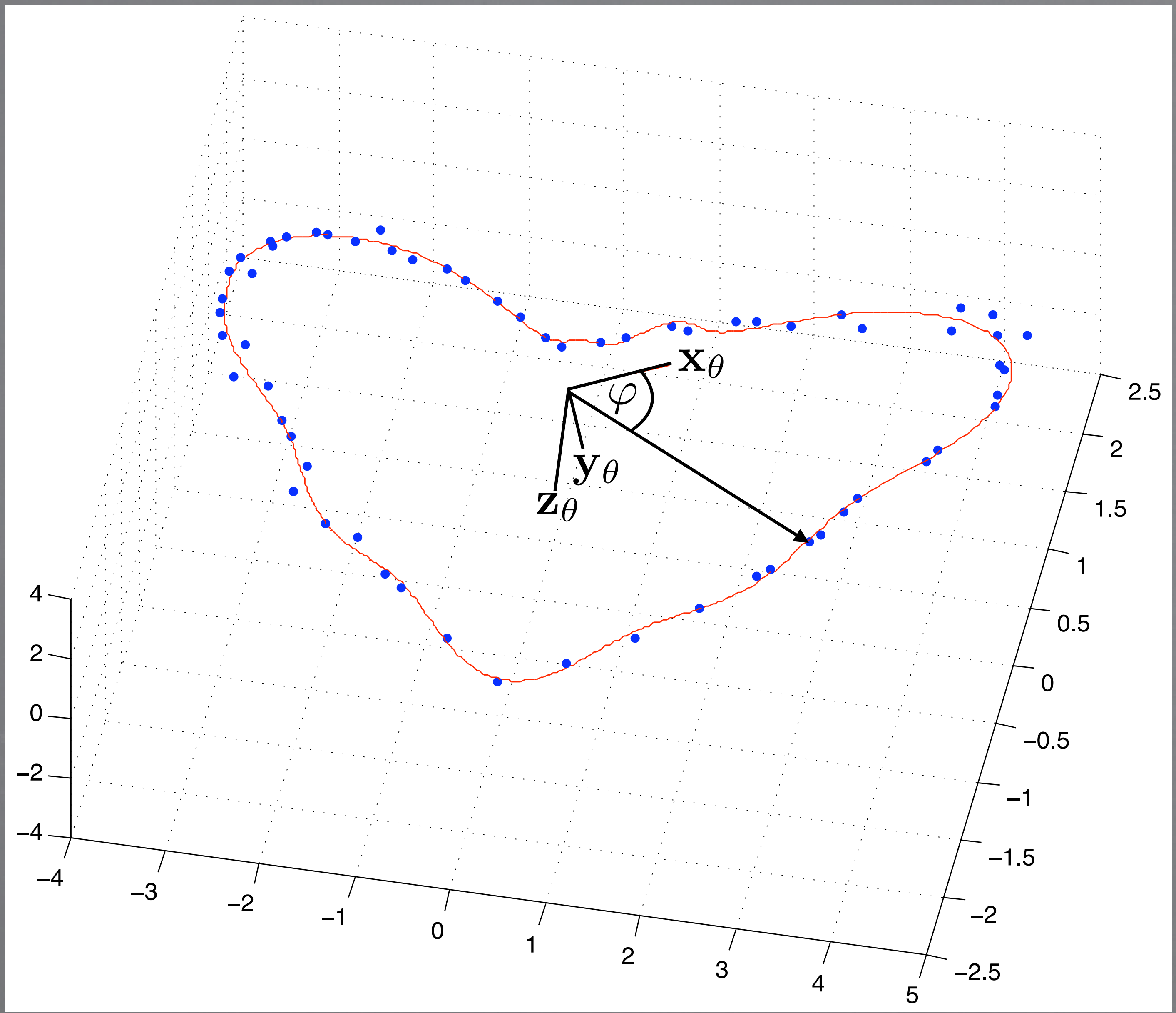
The “eigenpose” space

3-D low-dimensional subspaces by linear PCA





Action subspace embedding



Map data to cylindrical coordinate system

$$\mathbf{z}_\theta = \frac{\Sigma_i(\hat{\mathbf{x}}^i \times \hat{\mathbf{x}}^{i+1})}{\|\Sigma_i(\hat{\mathbf{x}}^i \times \hat{\mathbf{x}}^{i+1})\|}$$

Learn 1-D representation of motion in term of motion phase angle:

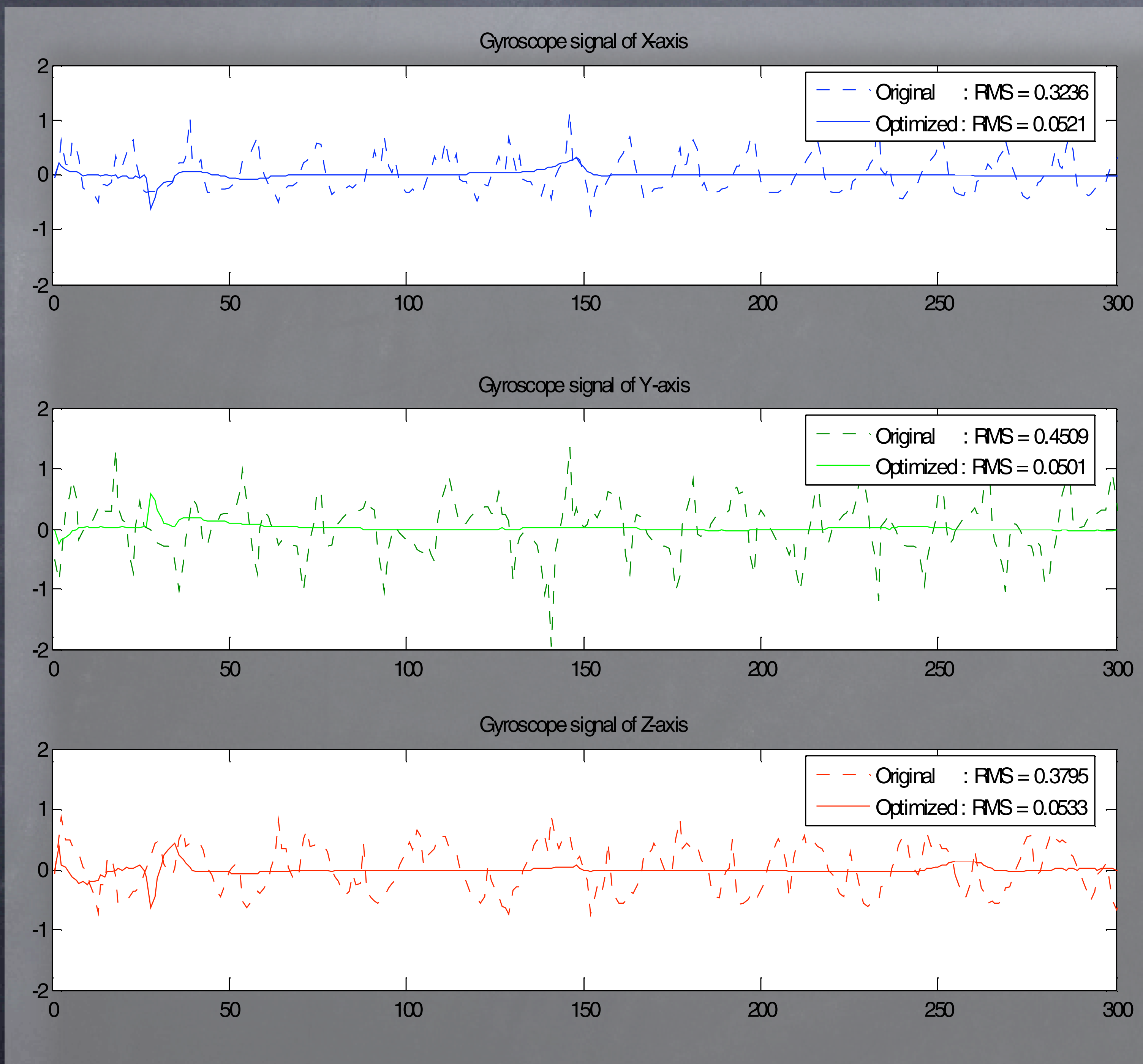
$$[r, h] = g(\varphi)$$

Presentation outline

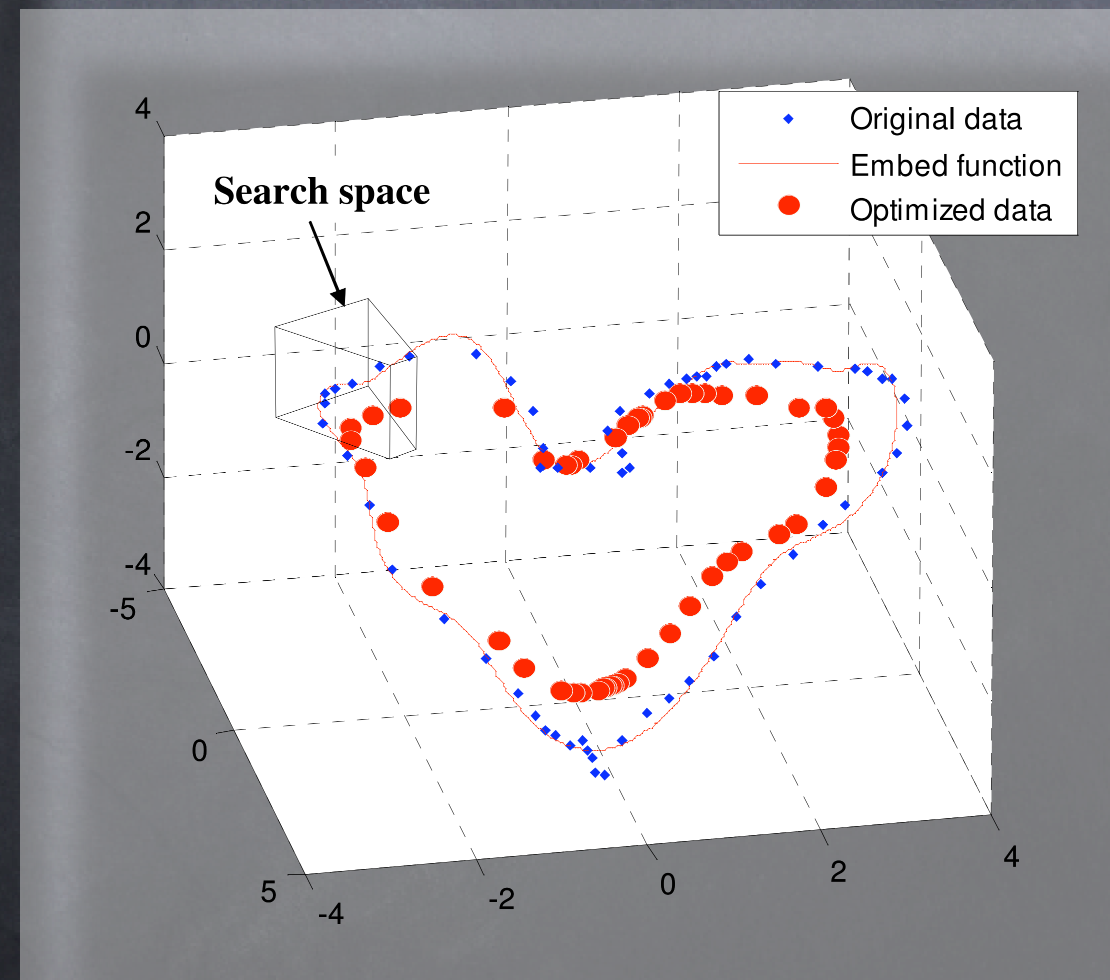
- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation

Optimization strategy

Gyroscope signals

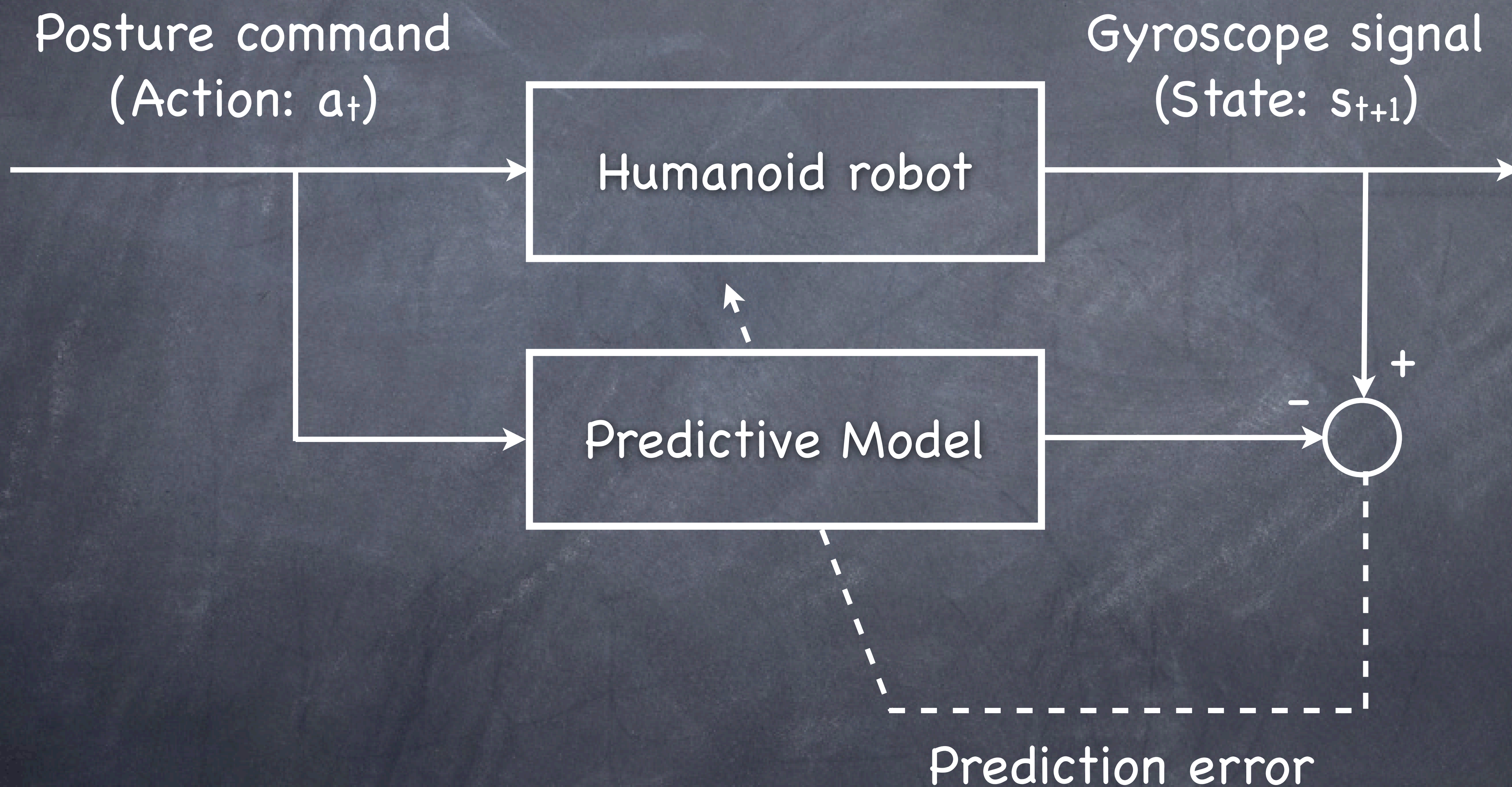


Optimized motion



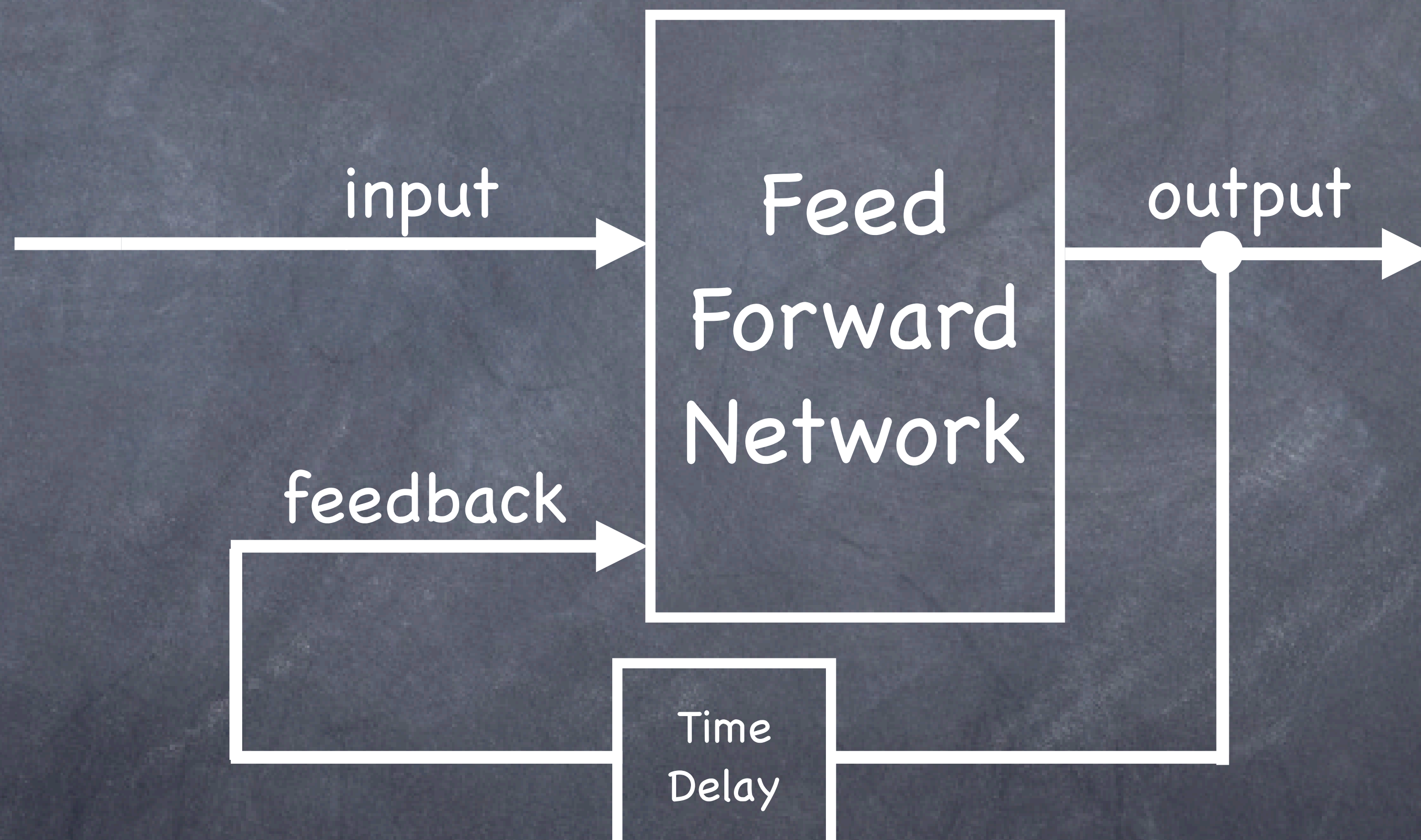
Learning the predictive model

$$s_{t+1} = F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n})$$



NARX model-predictor

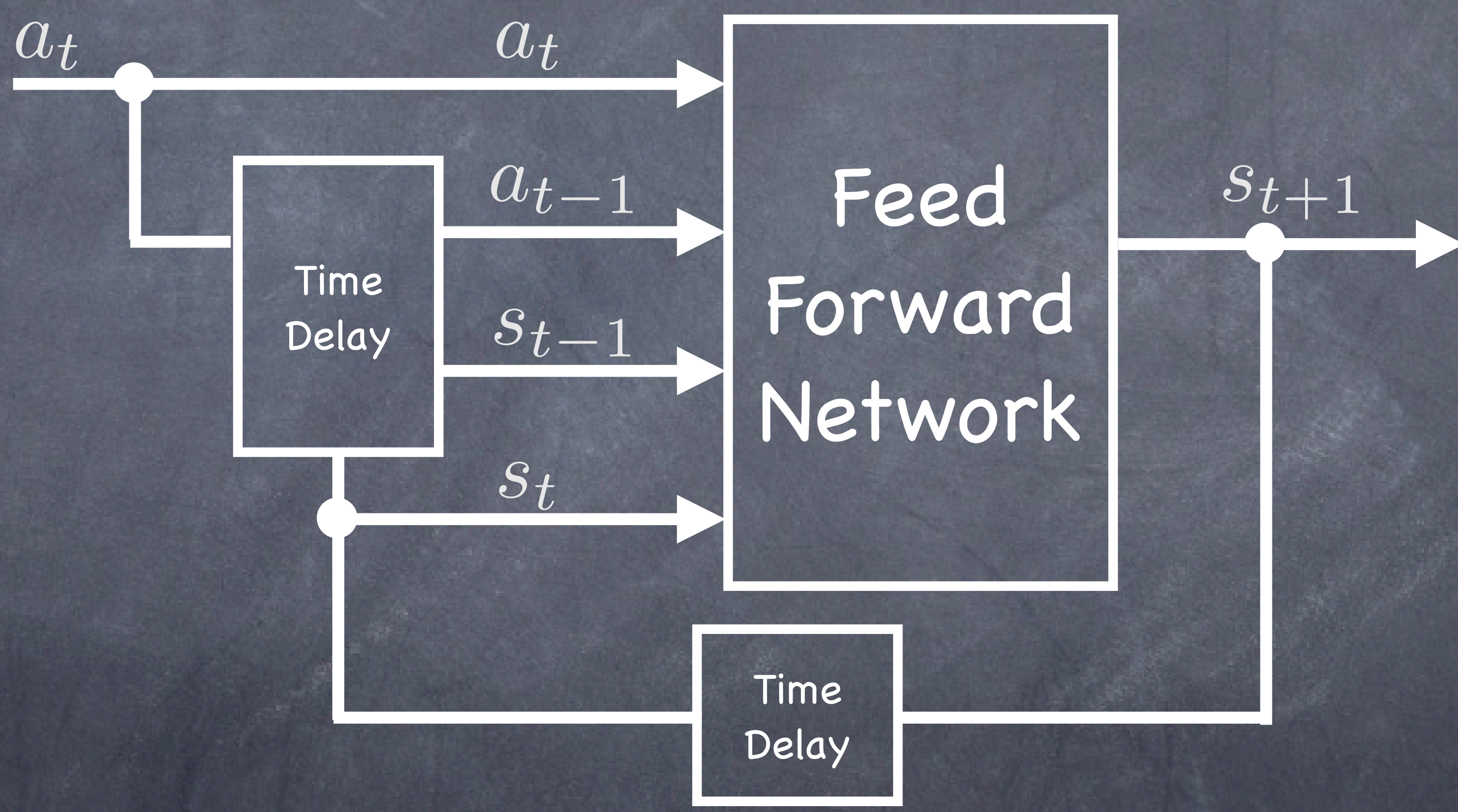
Nonlinear autoregressive network with exogenous inputs



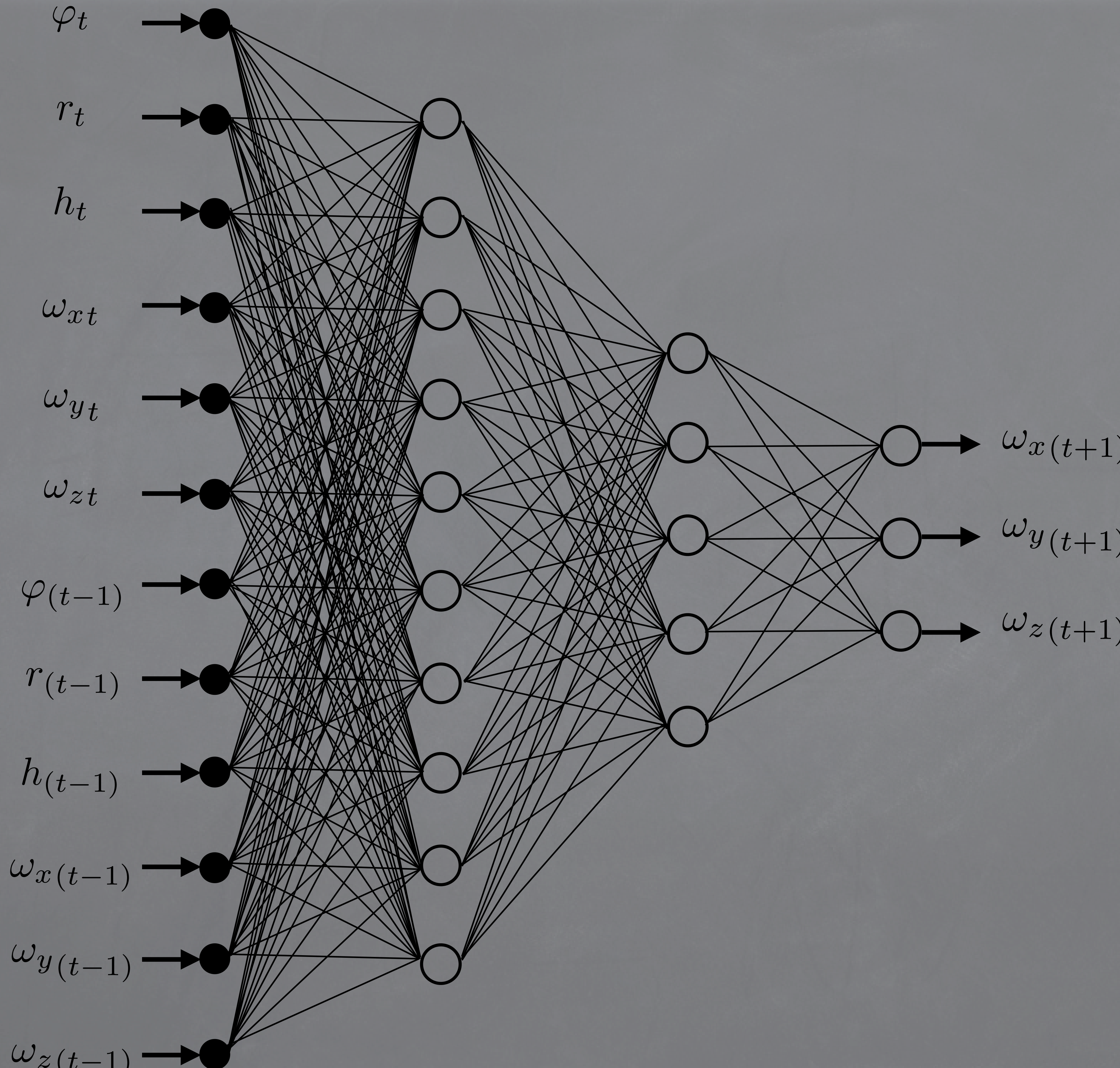
recurrent neural network

NARX model-predictor

Nonlinear autoregressive network with exogenous inputs

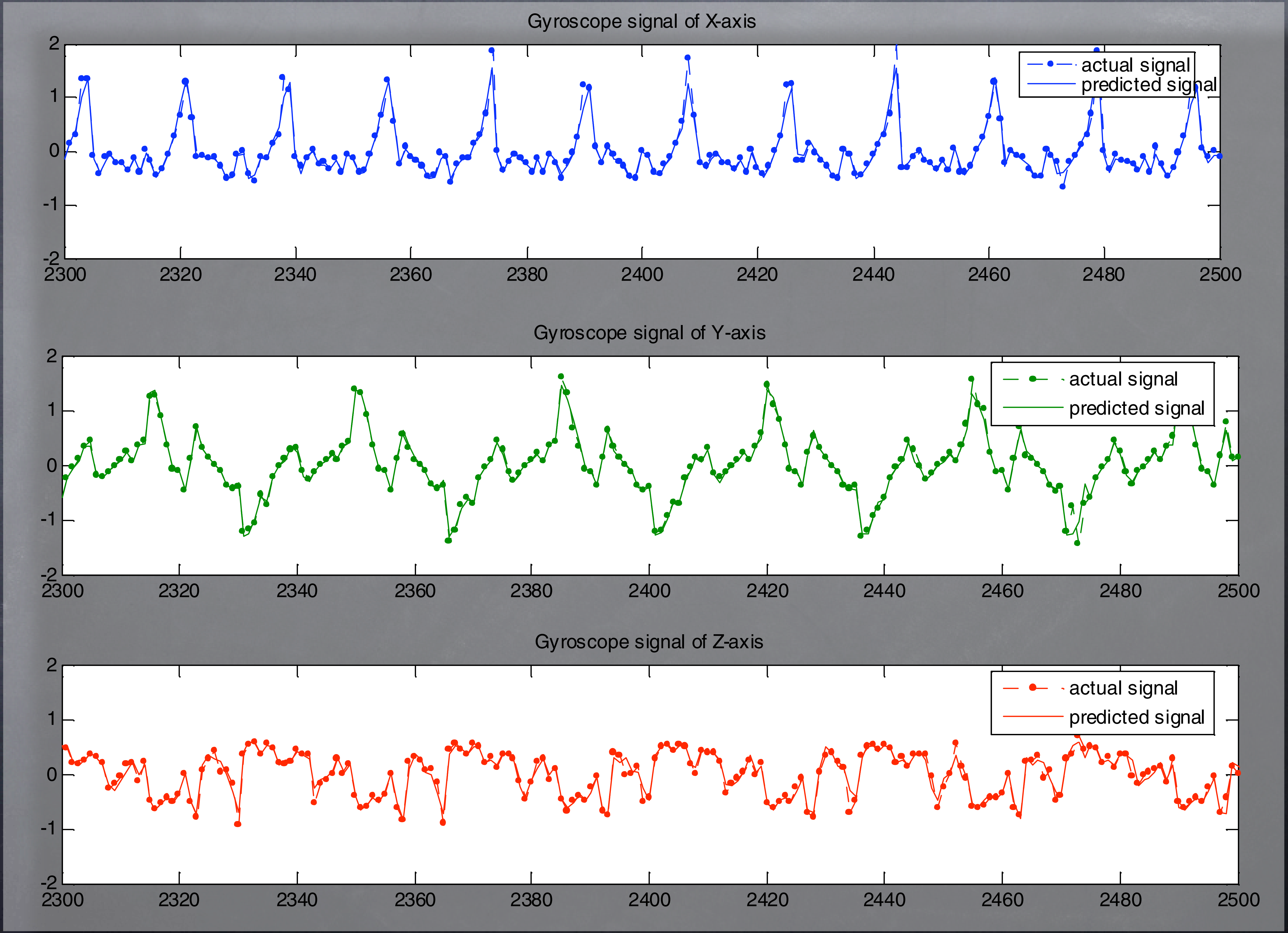


$$s_{t+1} = F(s_t, s_{t-1}, a_t, a_{t-1})$$



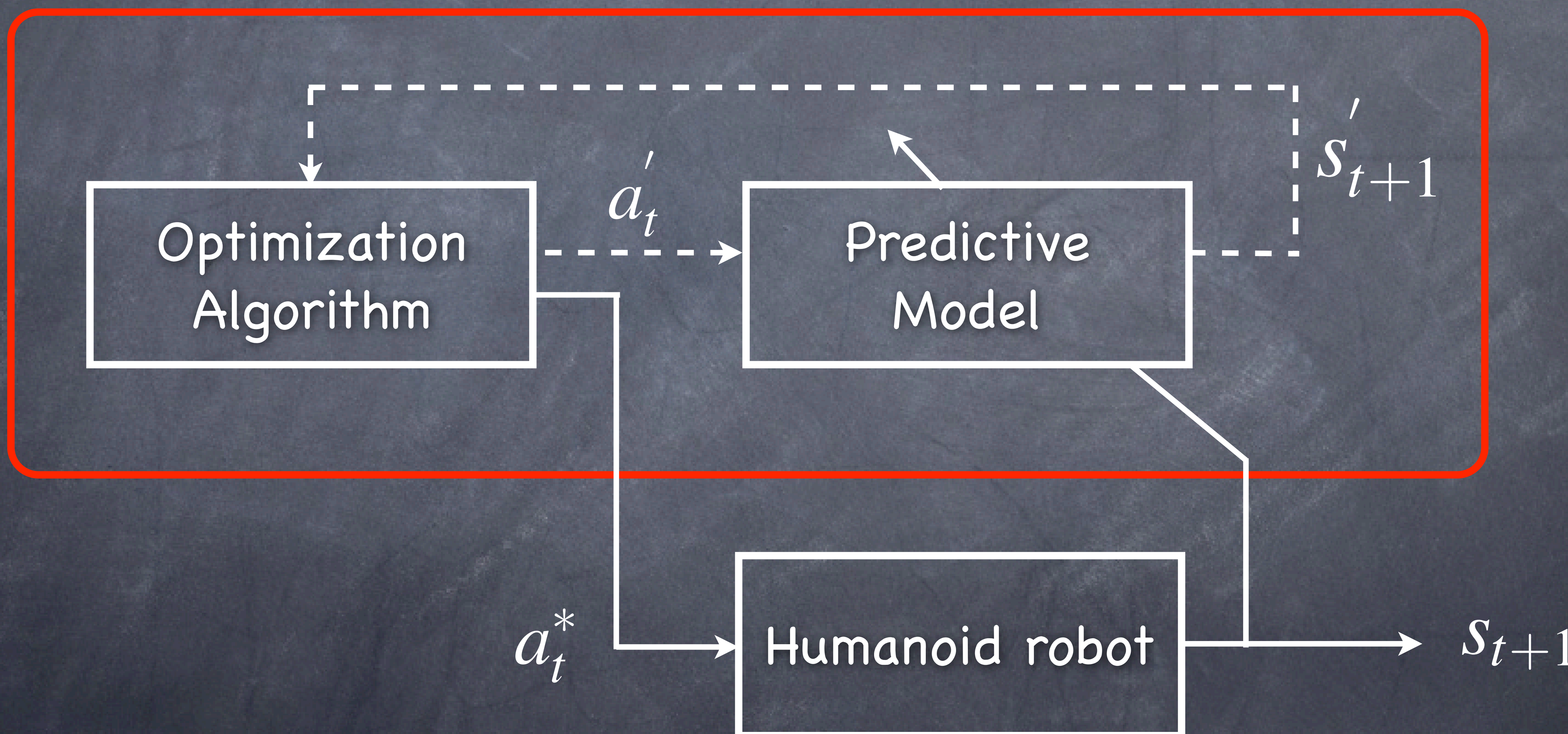
φ_t
r_t
h_t
ω_{xt}
ω_{yt}
ω_{zt}
$\varphi_{(t-1)}$
$r_{(t-1)}$
$h_{(t-1)}$
$\omega_{x(t-1)}$
$\omega_{y(t-1)}$
$\omega_{z(t-1)}$
$\omega_{x(t+1)}$
$\omega_{y(t+1)}$
$\omega_{z(t+1)}$

Gyroscope signals prediction



Predictive motion generator

$$a_t^* = \arg\min_{a_t} \Gamma(F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n}))$$

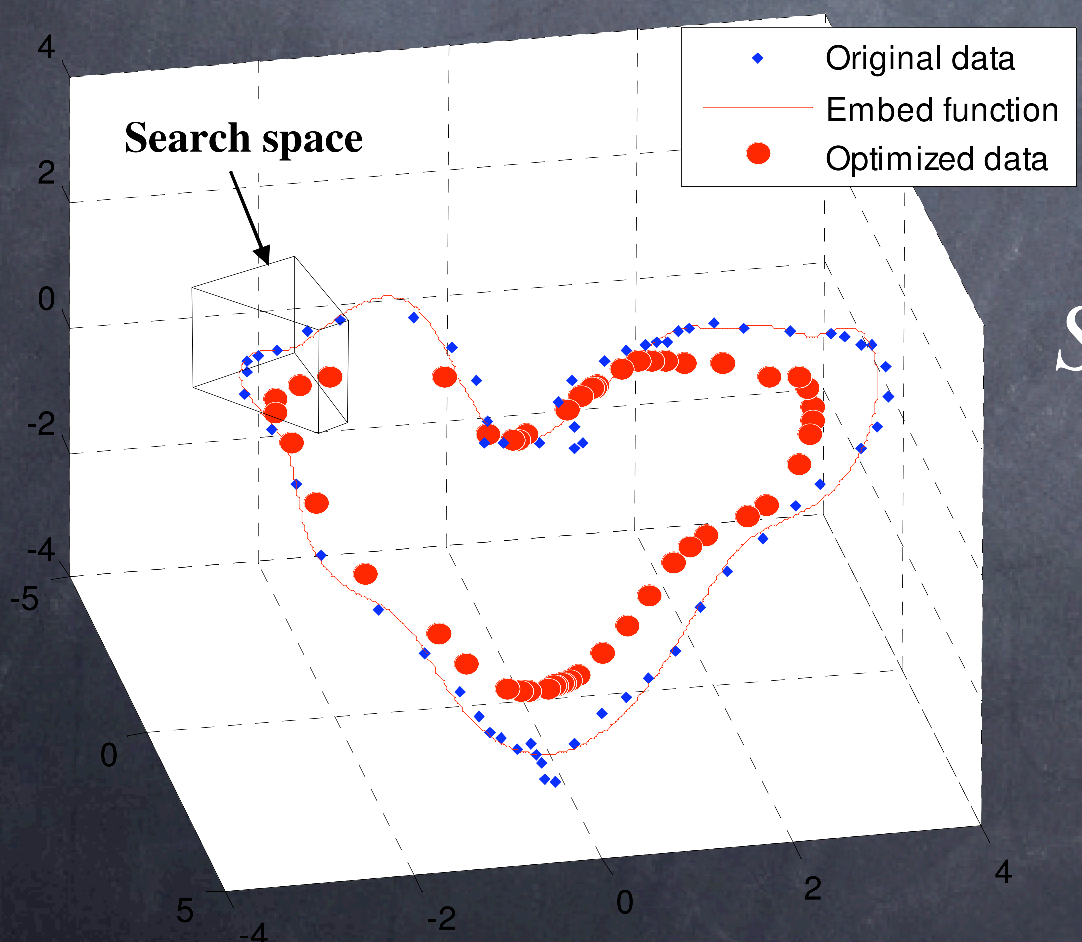


Maths details

$$a_t^* = \arg\min_{a_t} \Gamma(F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n}))$$

$$\Gamma(\omega) = \lambda_x \omega_x^2 + \lambda_y \omega_y^2 + \lambda_z \omega_z^2$$

$$\chi_t^* = \arg\min_{\chi_t \in S} \Gamma(F(\omega_t, \omega_{t-1}, \chi_t, \chi_{t-1}))$$



$$S = \begin{bmatrix} \varphi_s \\ r_s \\ h_s \end{bmatrix}$$

$$\varphi_{t-1} < \varphi_s \leq \varphi_{t-1} + \varepsilon_\varphi$$

$$r_a - \varepsilon_r \leq r_s \leq r_a + \varepsilon_r$$

$$h_a - \varepsilon_h \leq h_s \leq h_a + \varepsilon_h$$

$$0 < \varepsilon_\varphi < 2\pi$$

$$[r_a, h_a] = g(\varphi_s)$$

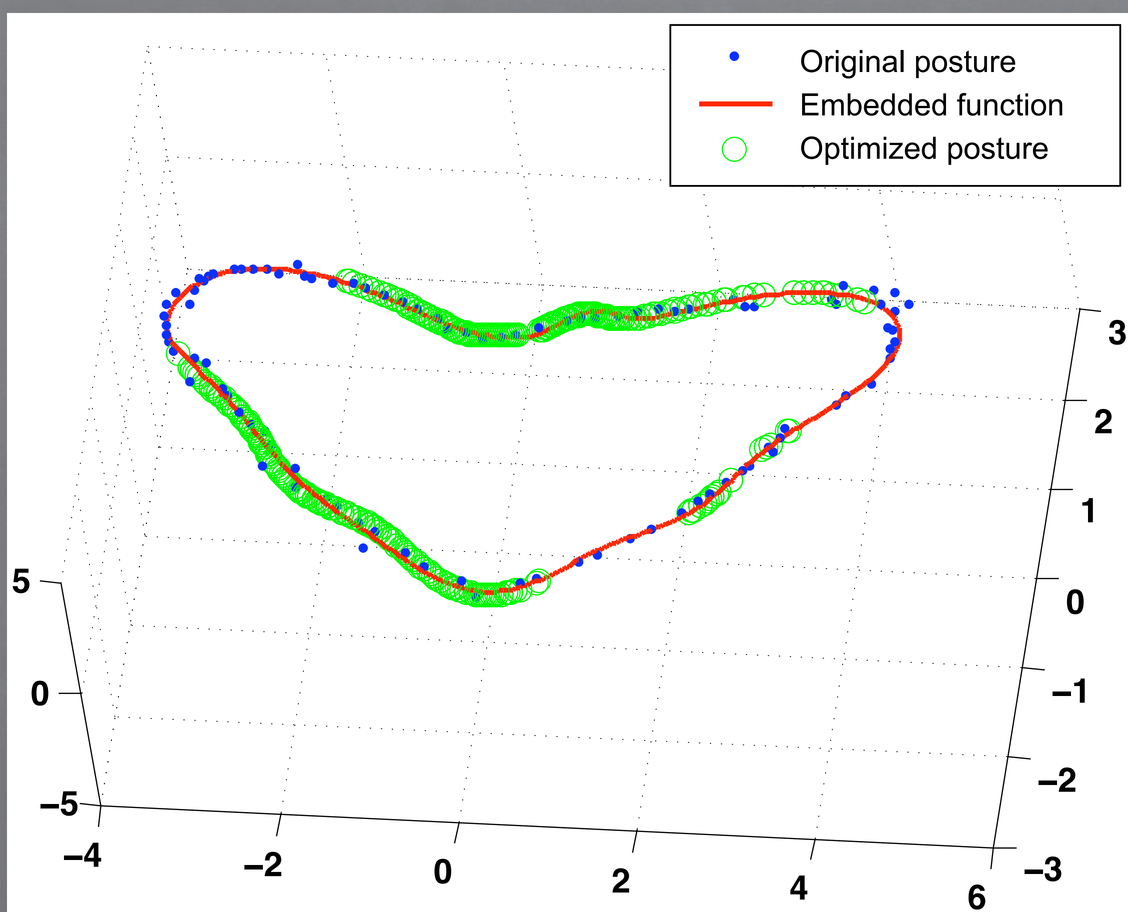
Presentation outline

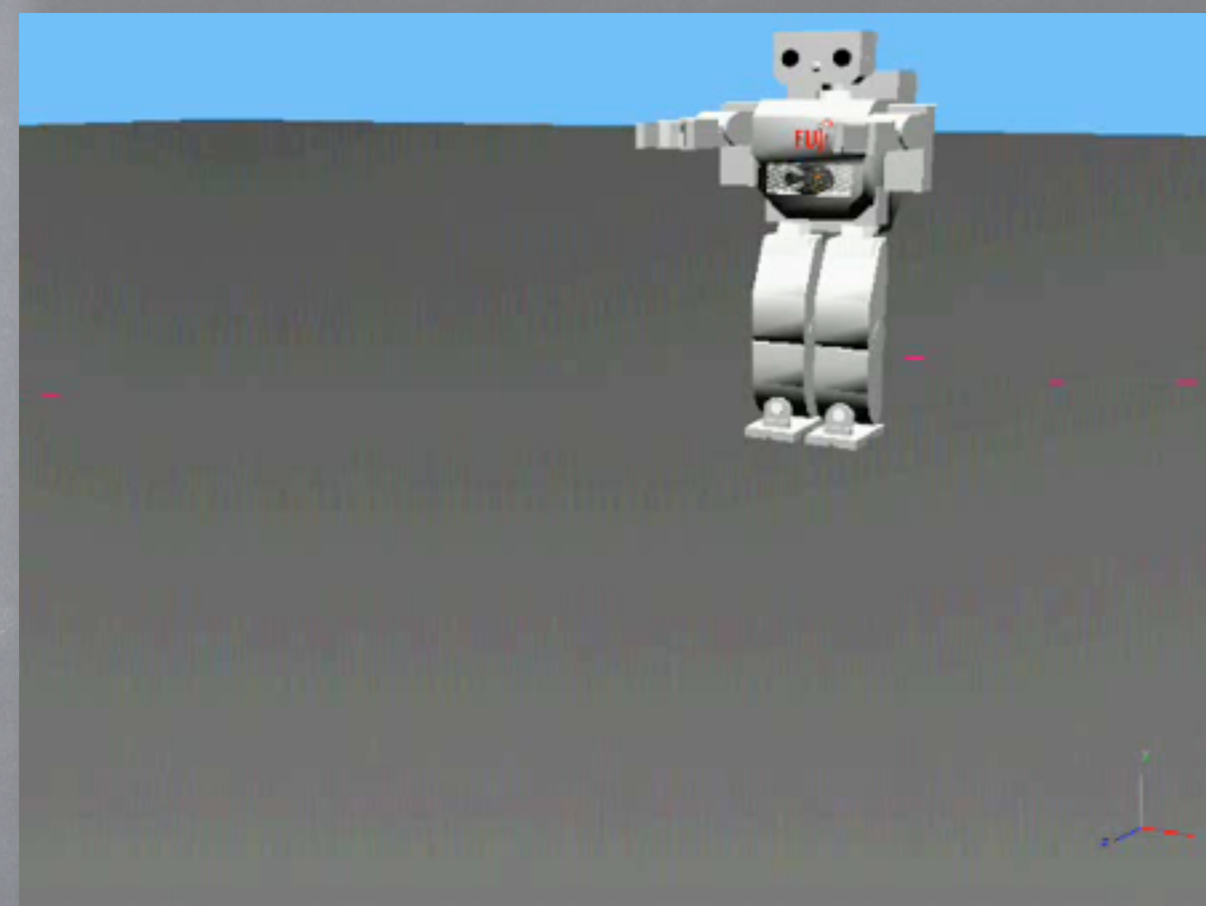
- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation

Motion-phase optimization

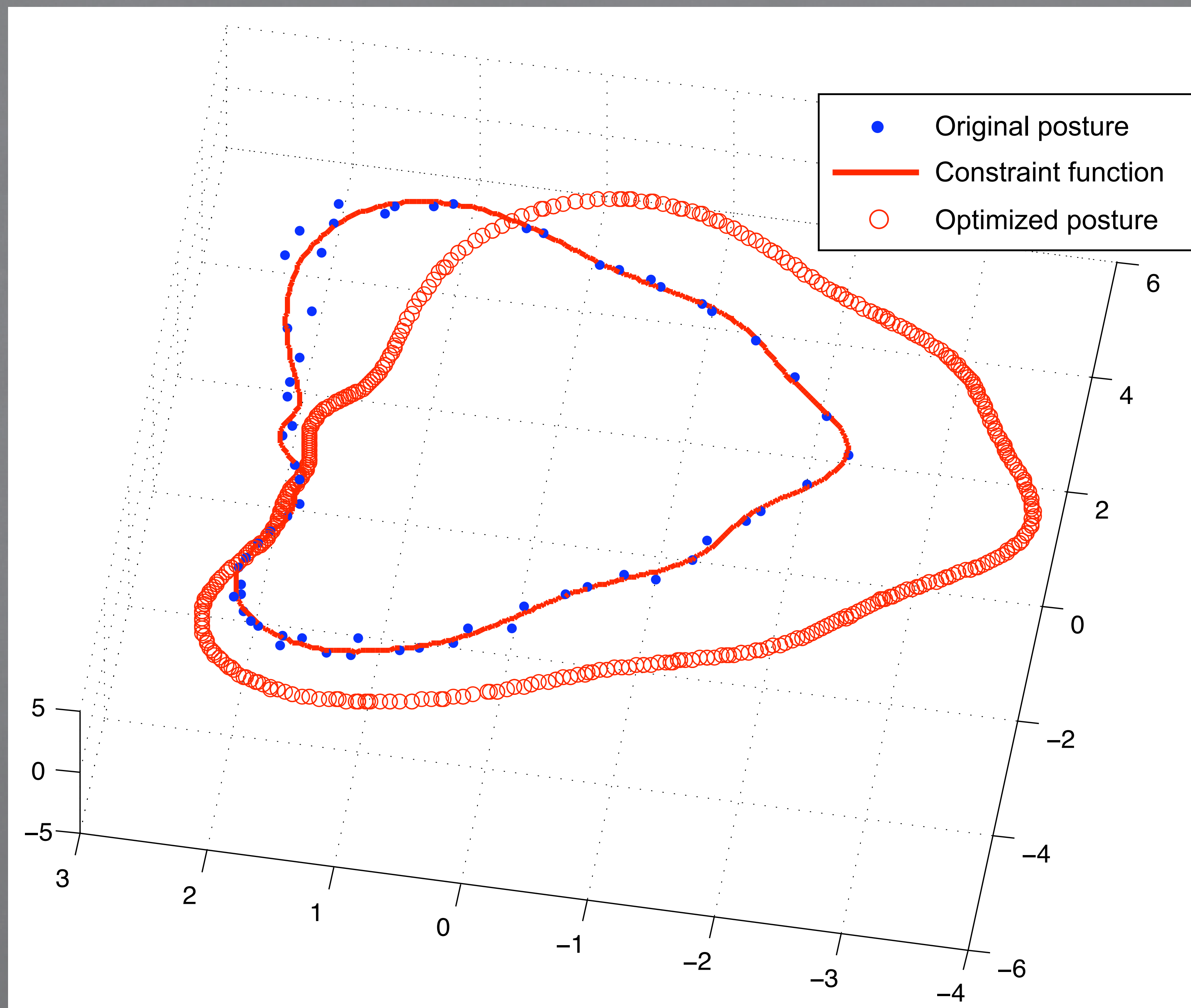
$$[r, h] = g(\varphi)$$

$$\varphi_t^* = \arg\min_{\varphi_t} \Gamma(F(\omega_t, \omega_{t-1}, \varphi_t, \varphi_{t-1}))$$

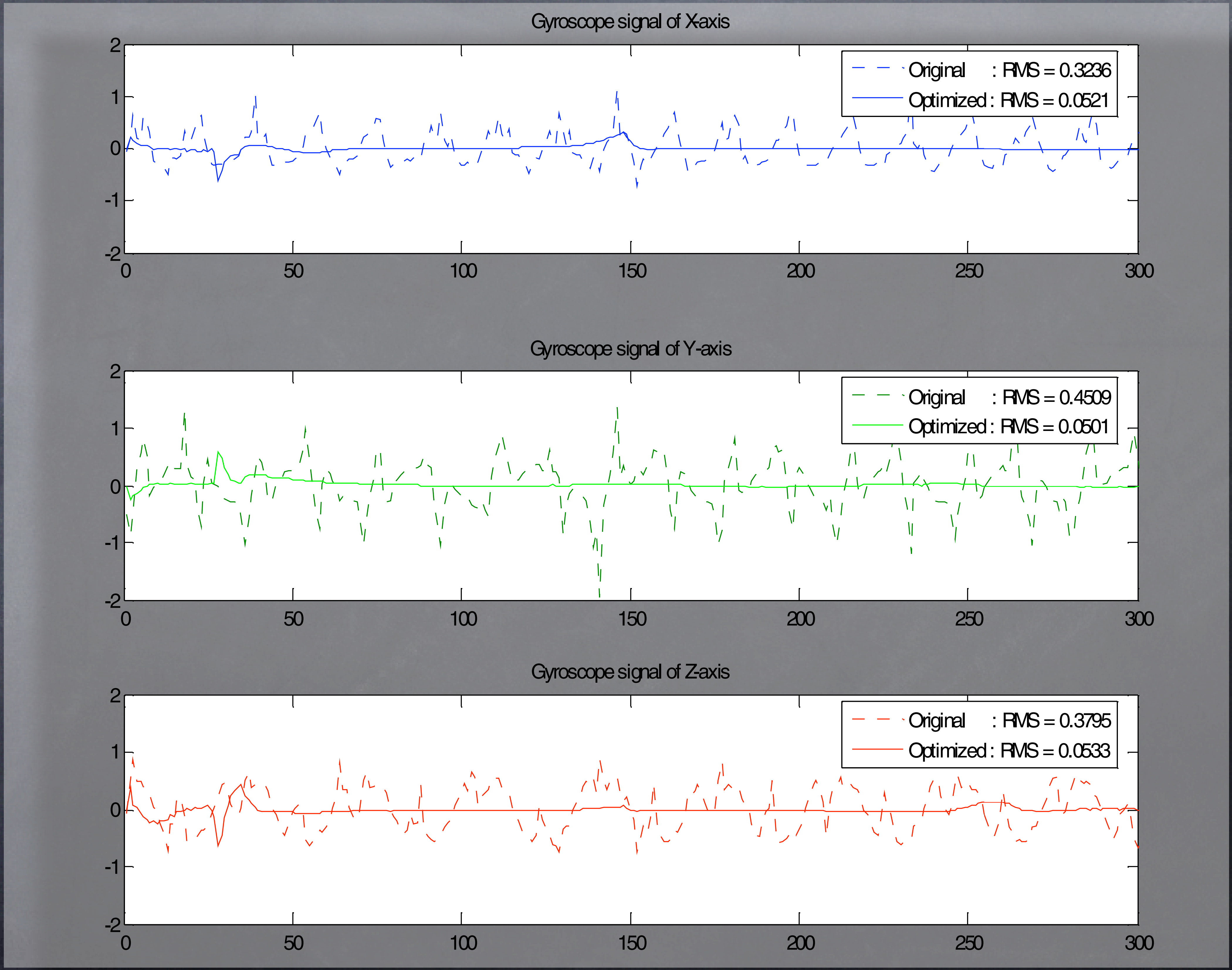




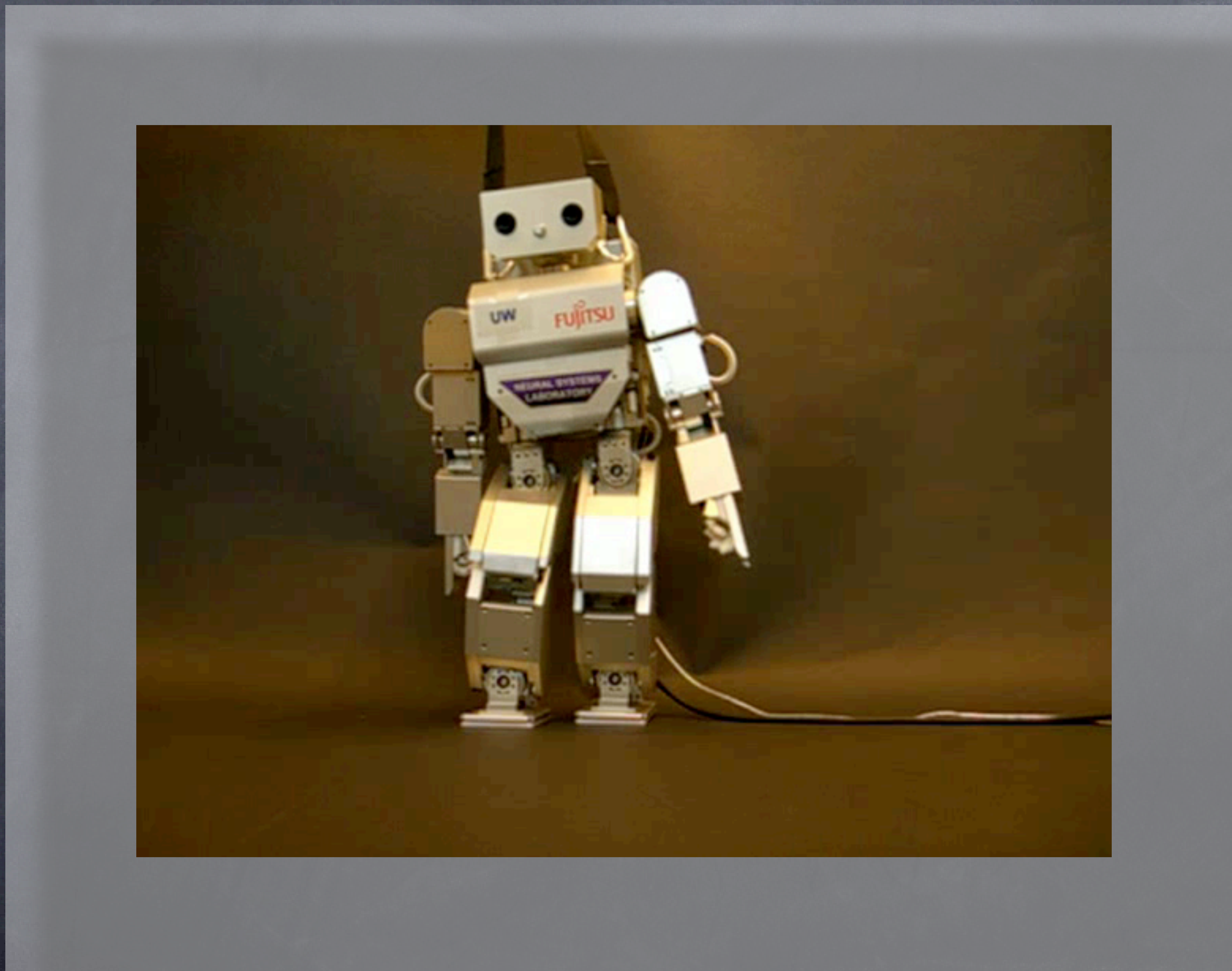
3-D Eigenposes optimization result



3-D Eigenposes optimization result



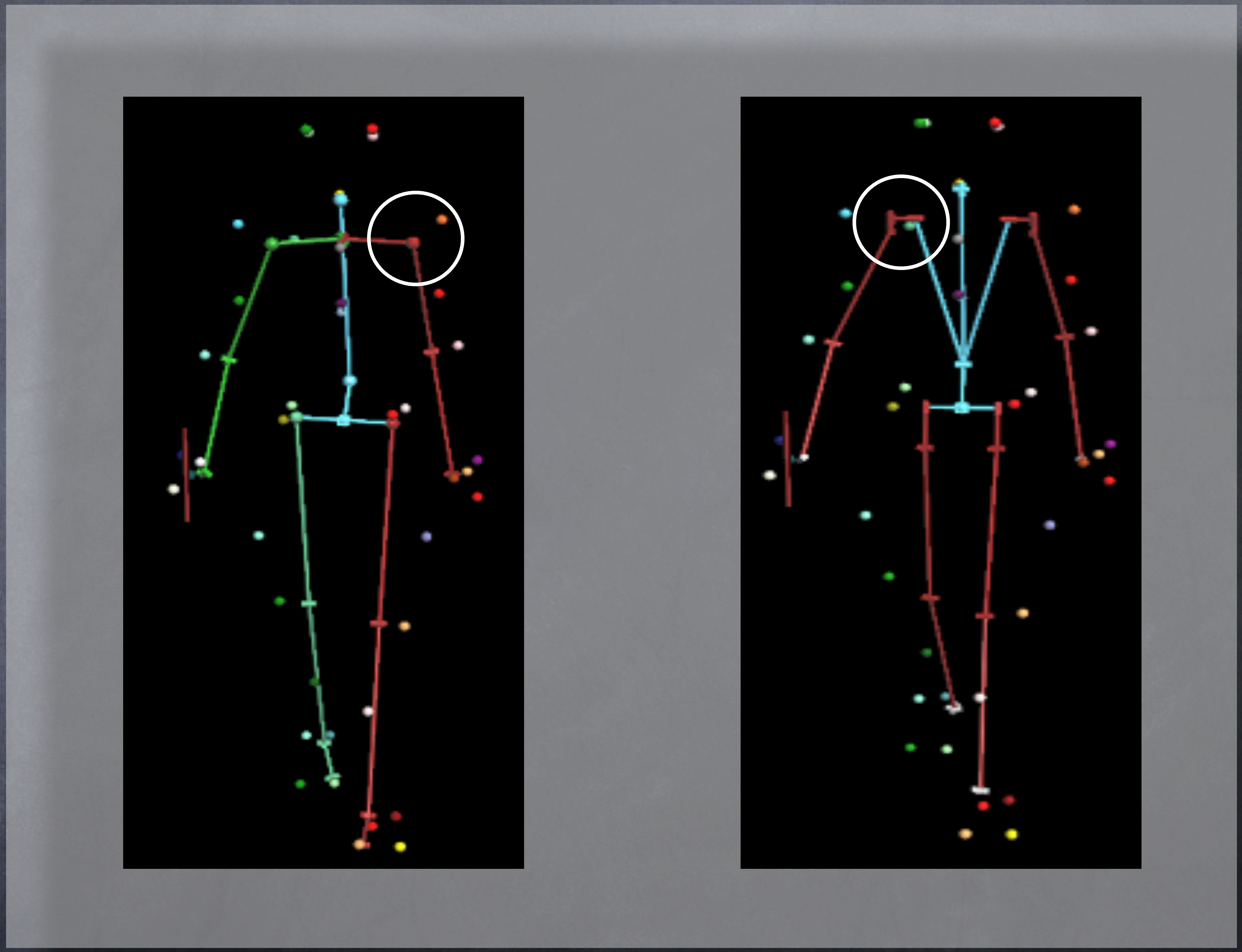
3-D Eigenposes optimization result



Presentation outline

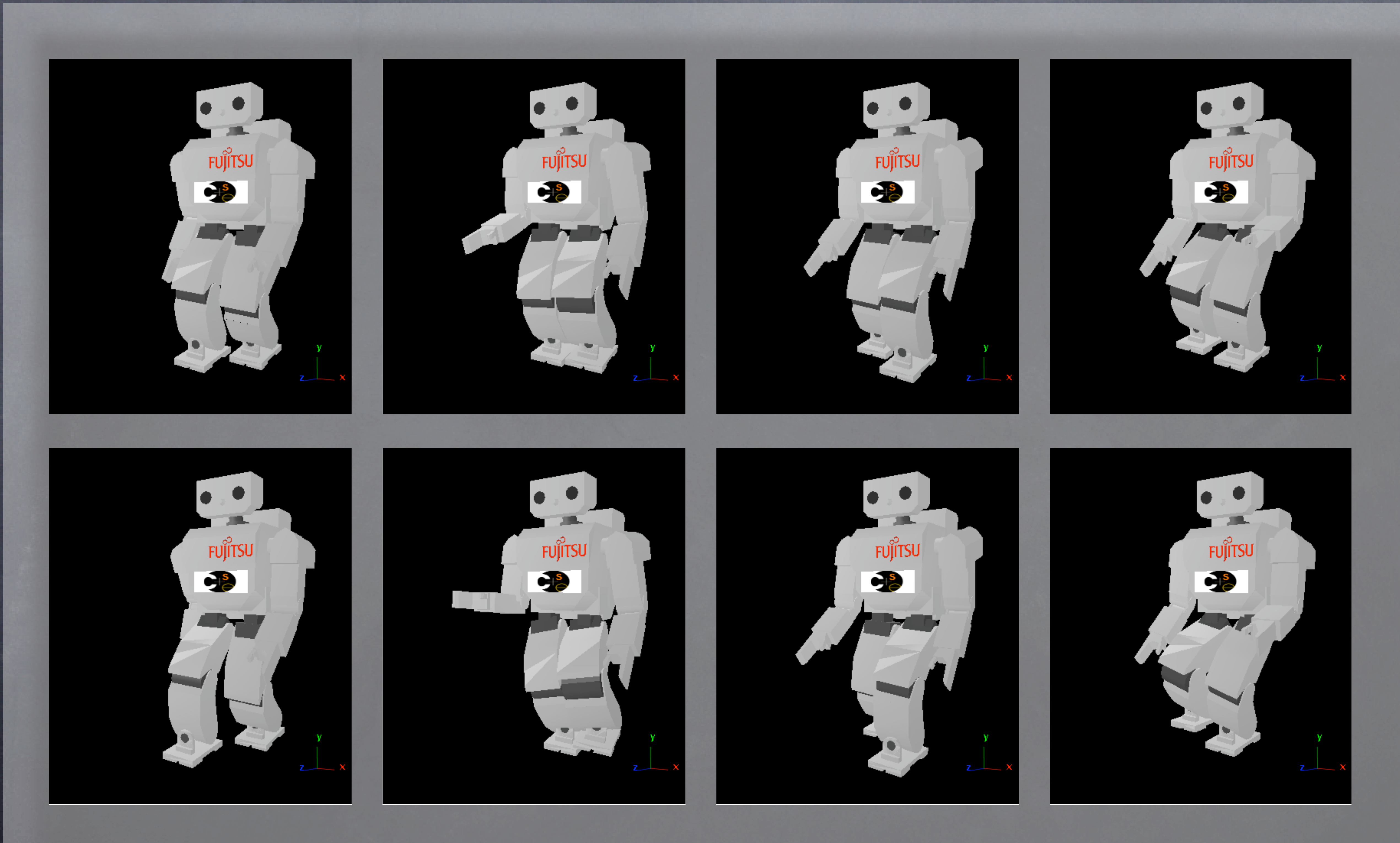
- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation

Human motion capture mapping

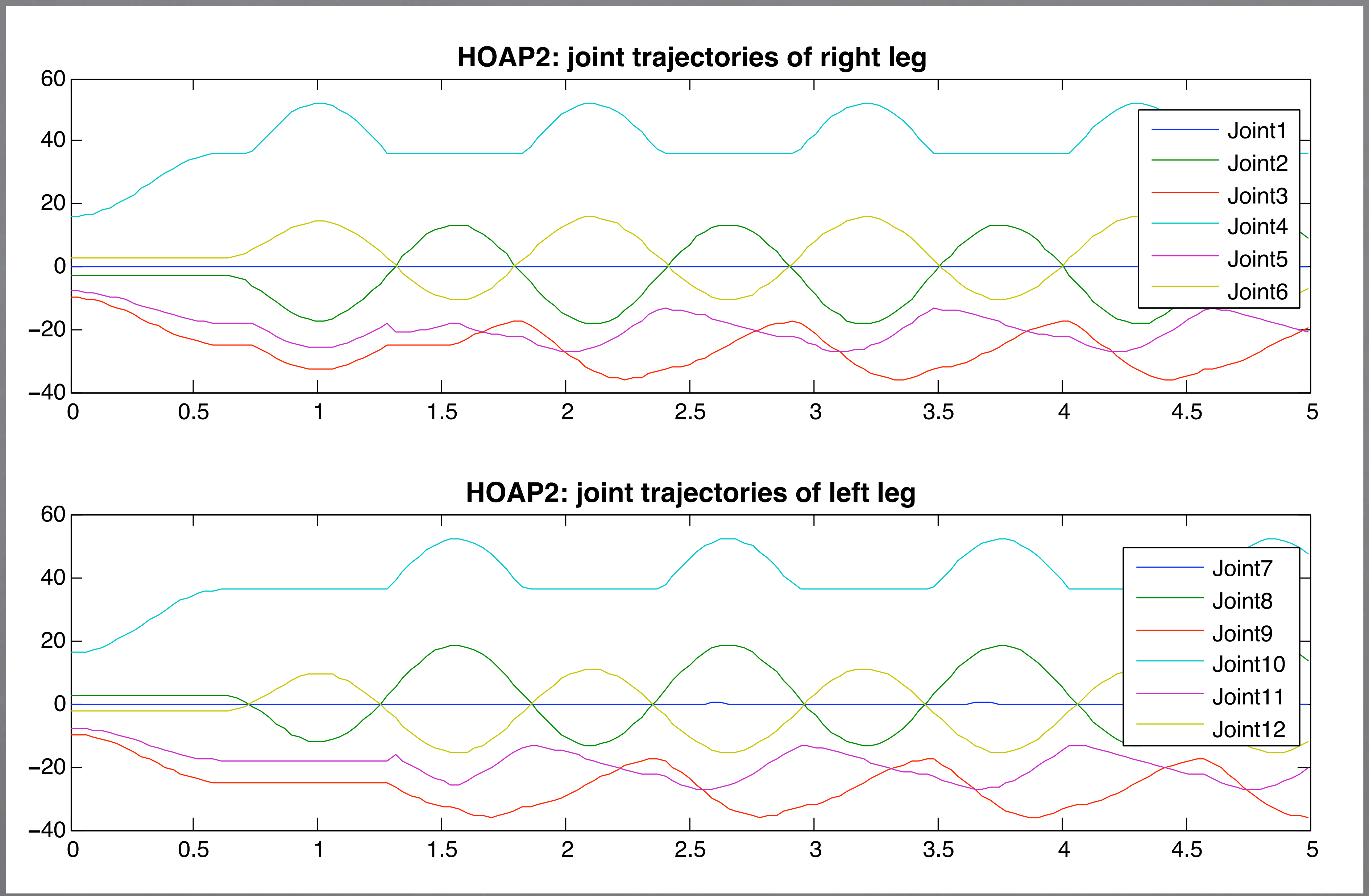


Human skeleton　　　　Robot skeleton

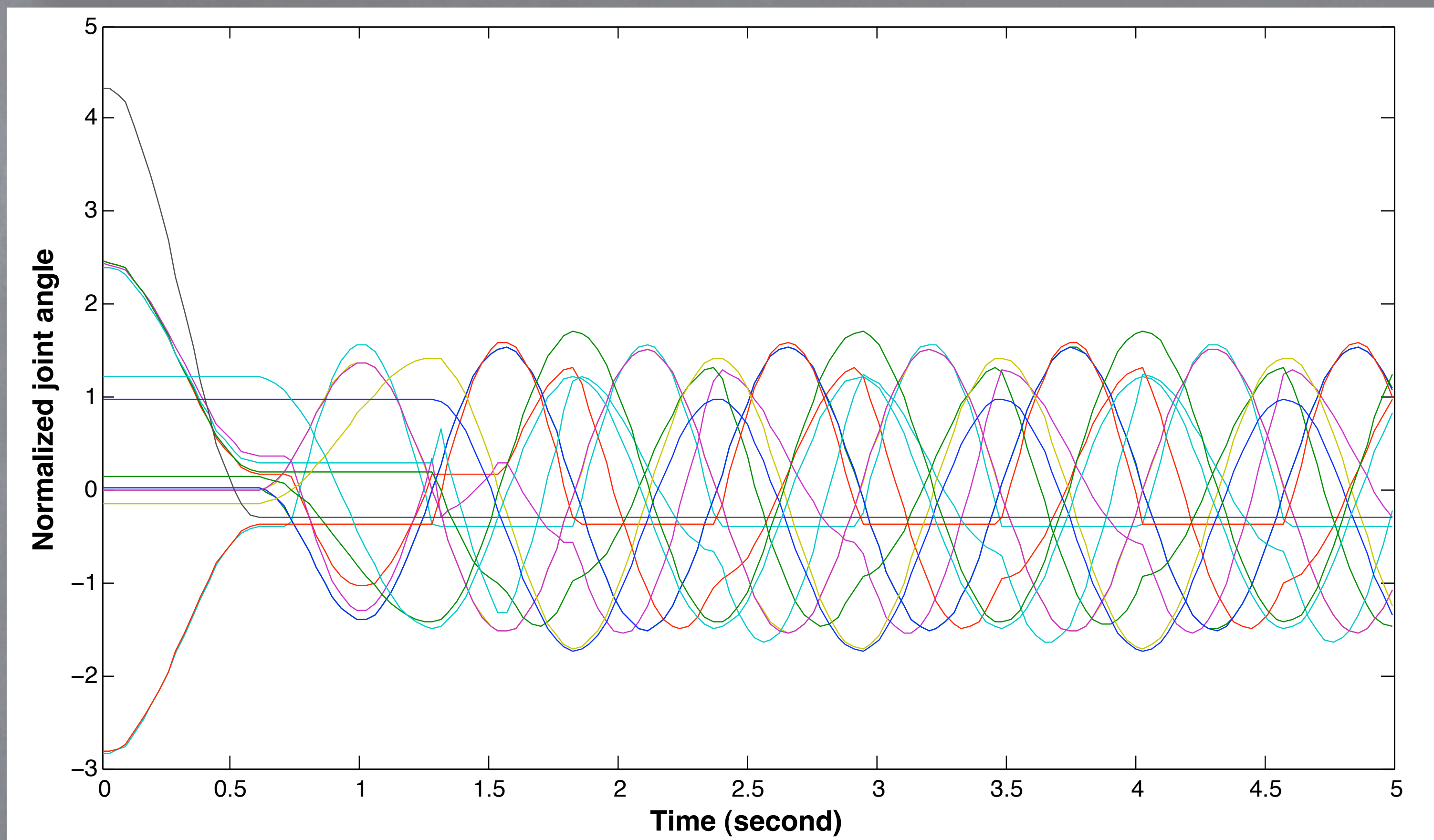
Motion scaling

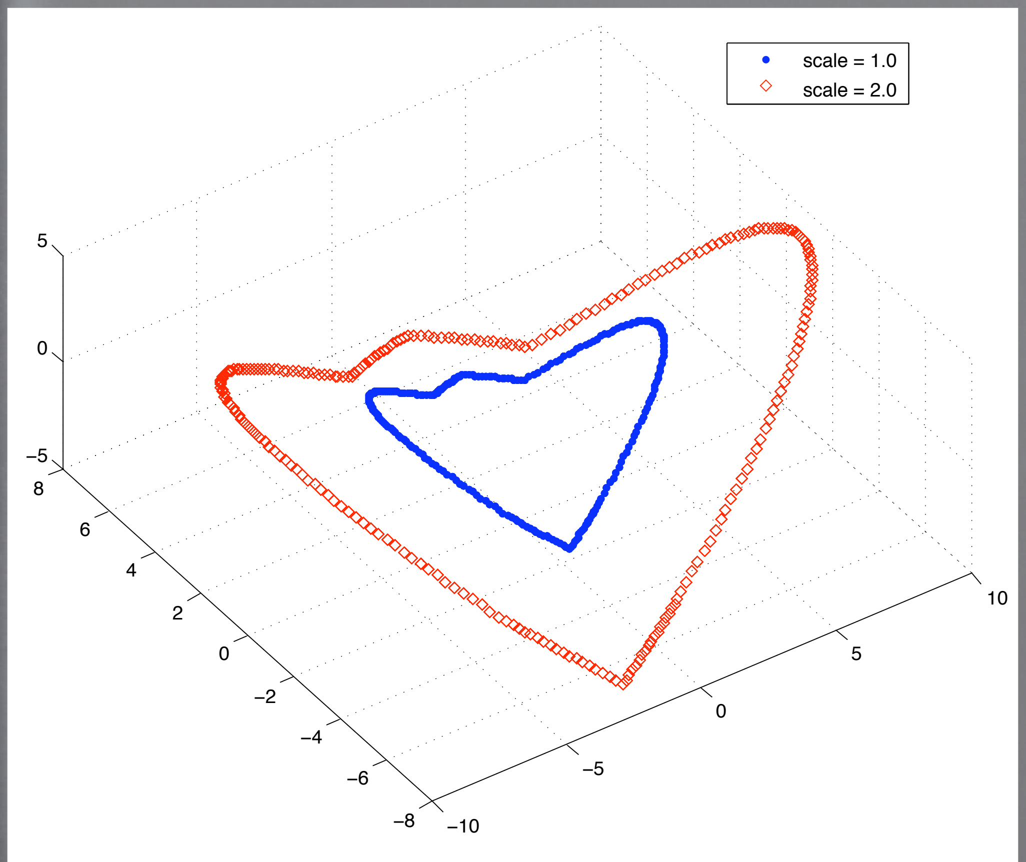


Joint trajectories



Action subspace scaling

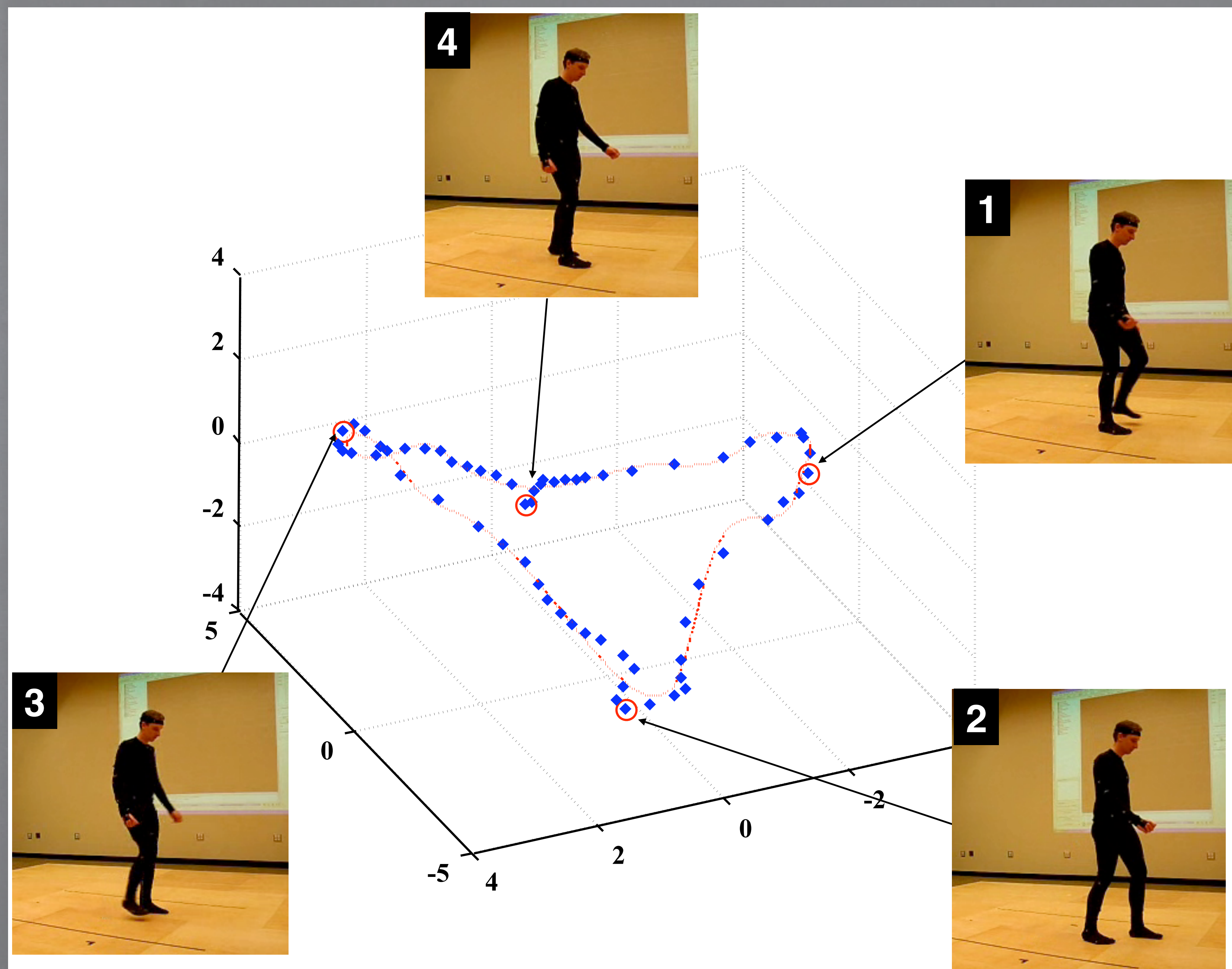




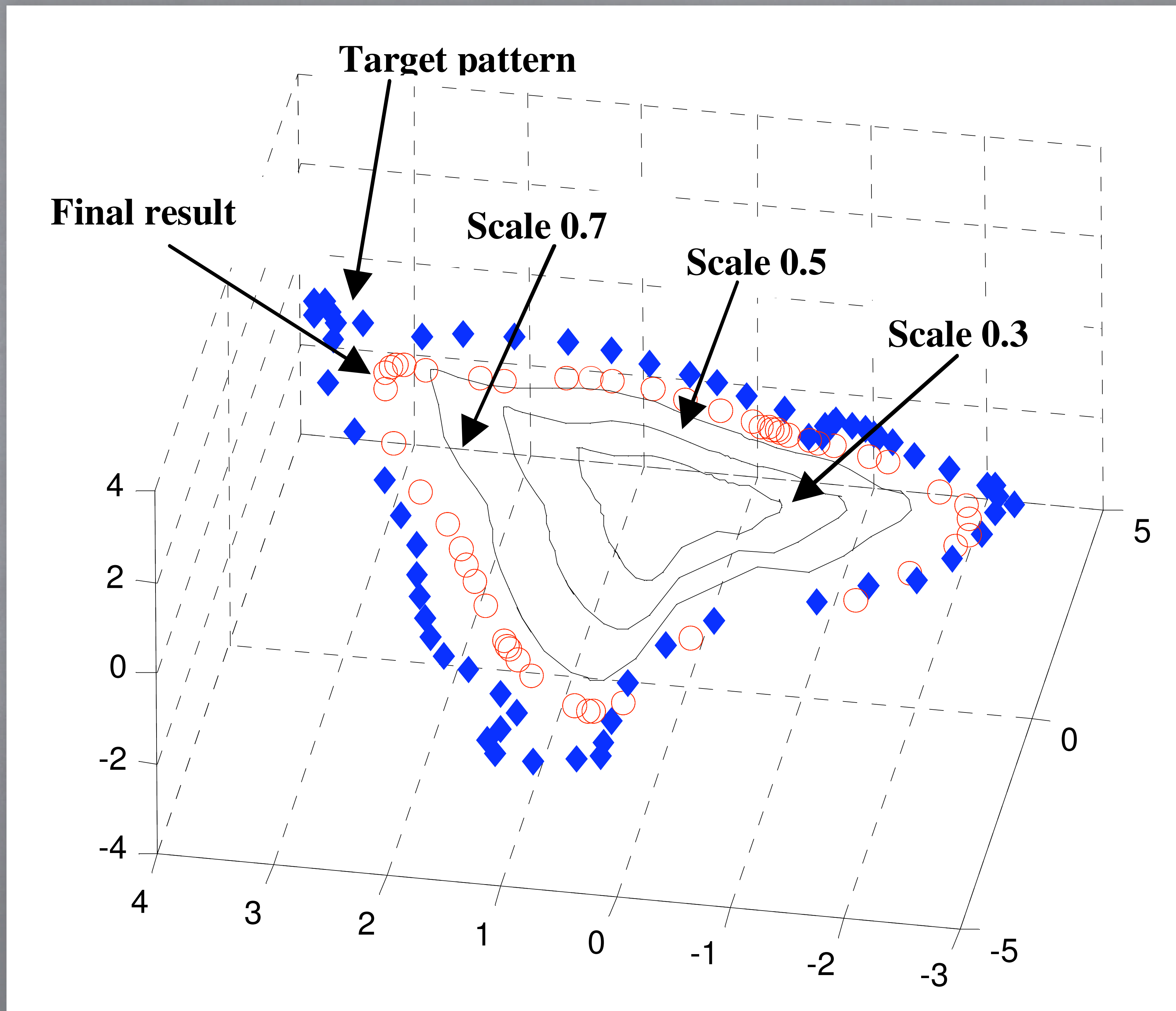
Normalized joint data
mean = 0
standard deviation = 1

Action subspace scaling

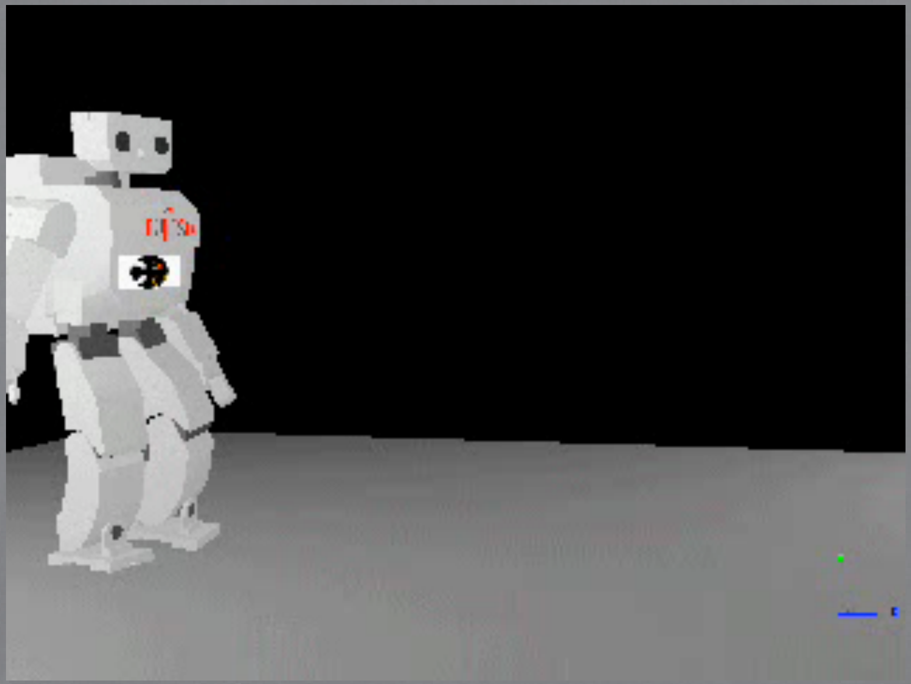
Imitate a human walking gait



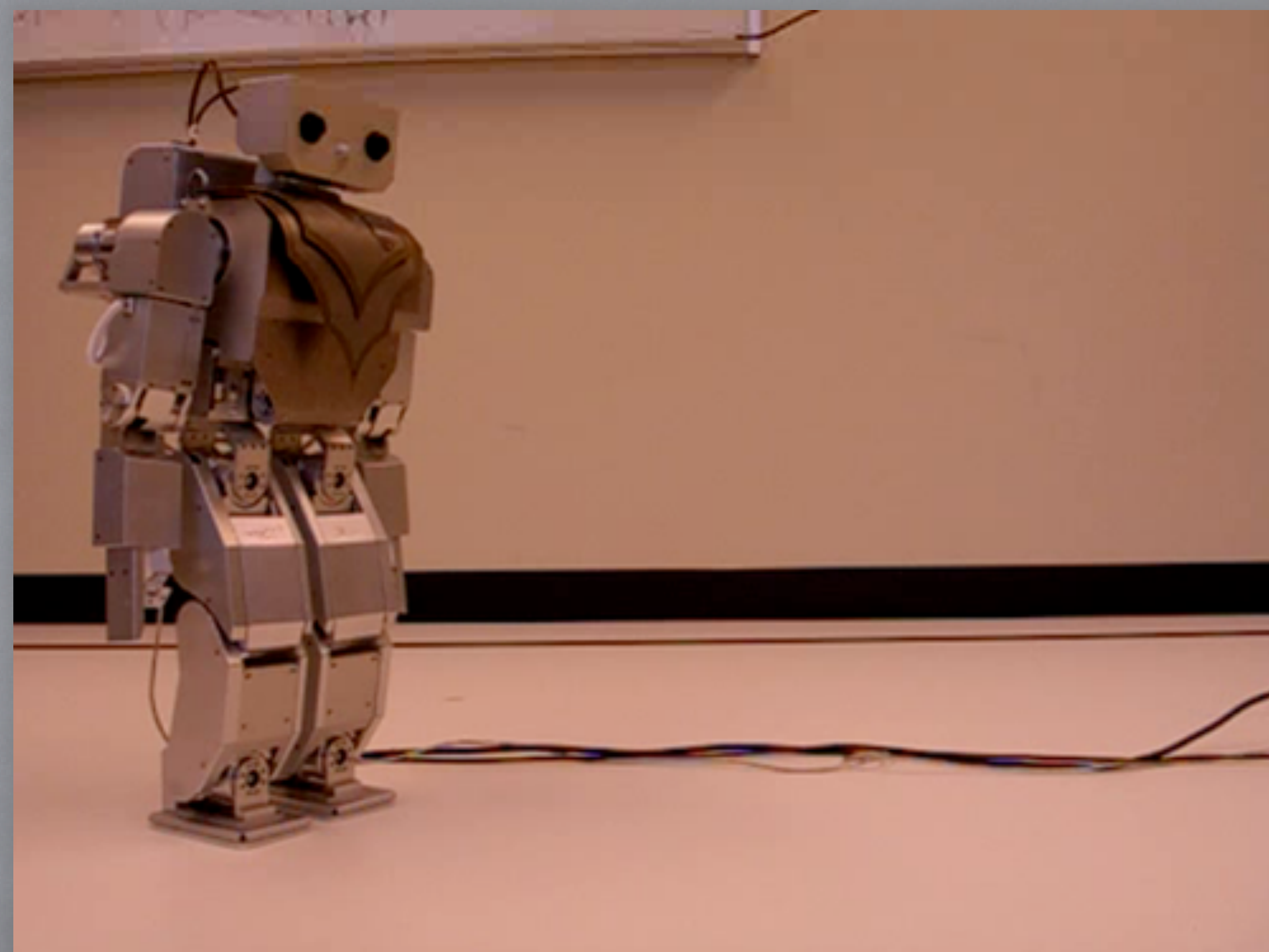
Imitate a human walking gait



Walking by imitation results



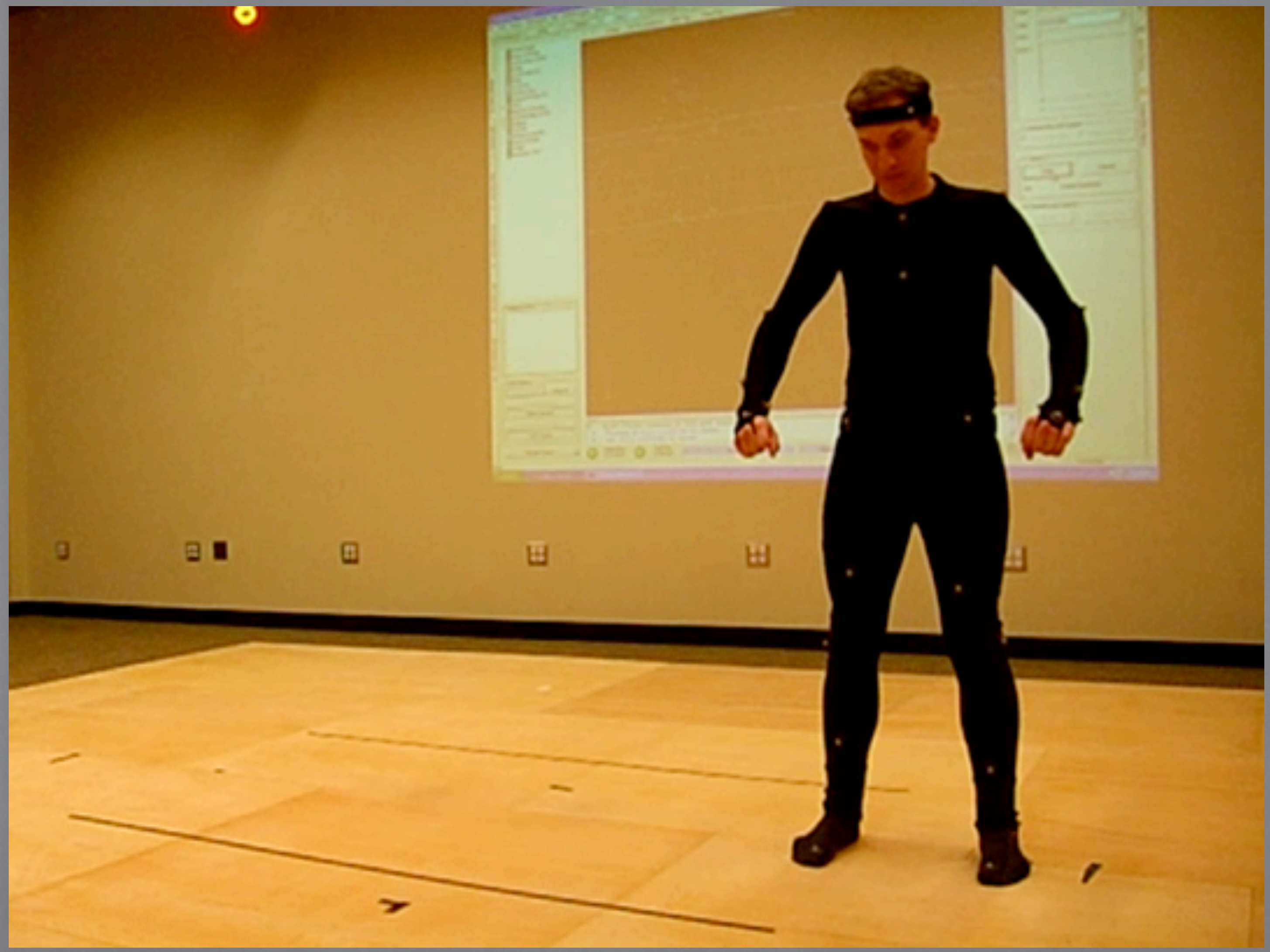
Walking by imitation results



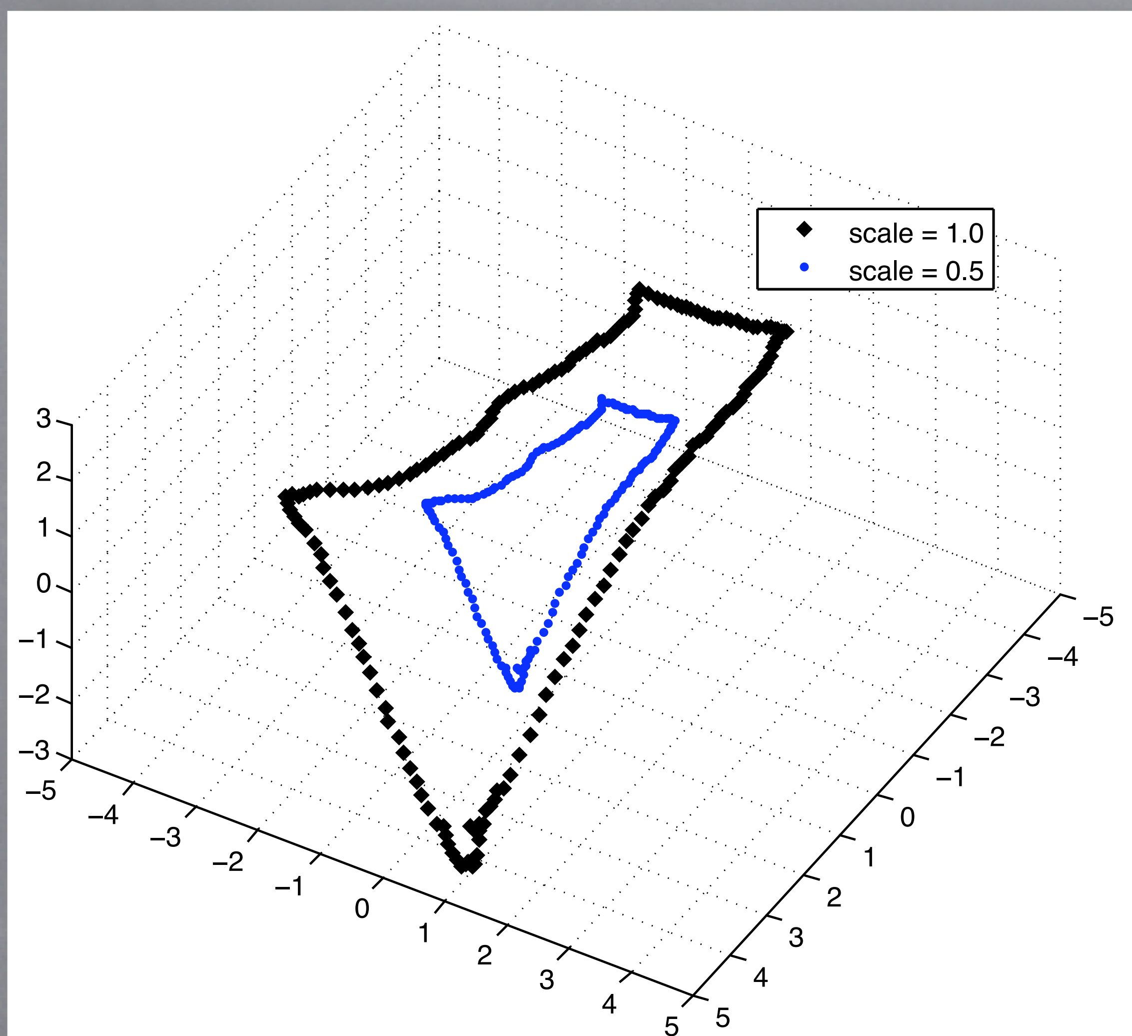
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation

Human sidestep motion

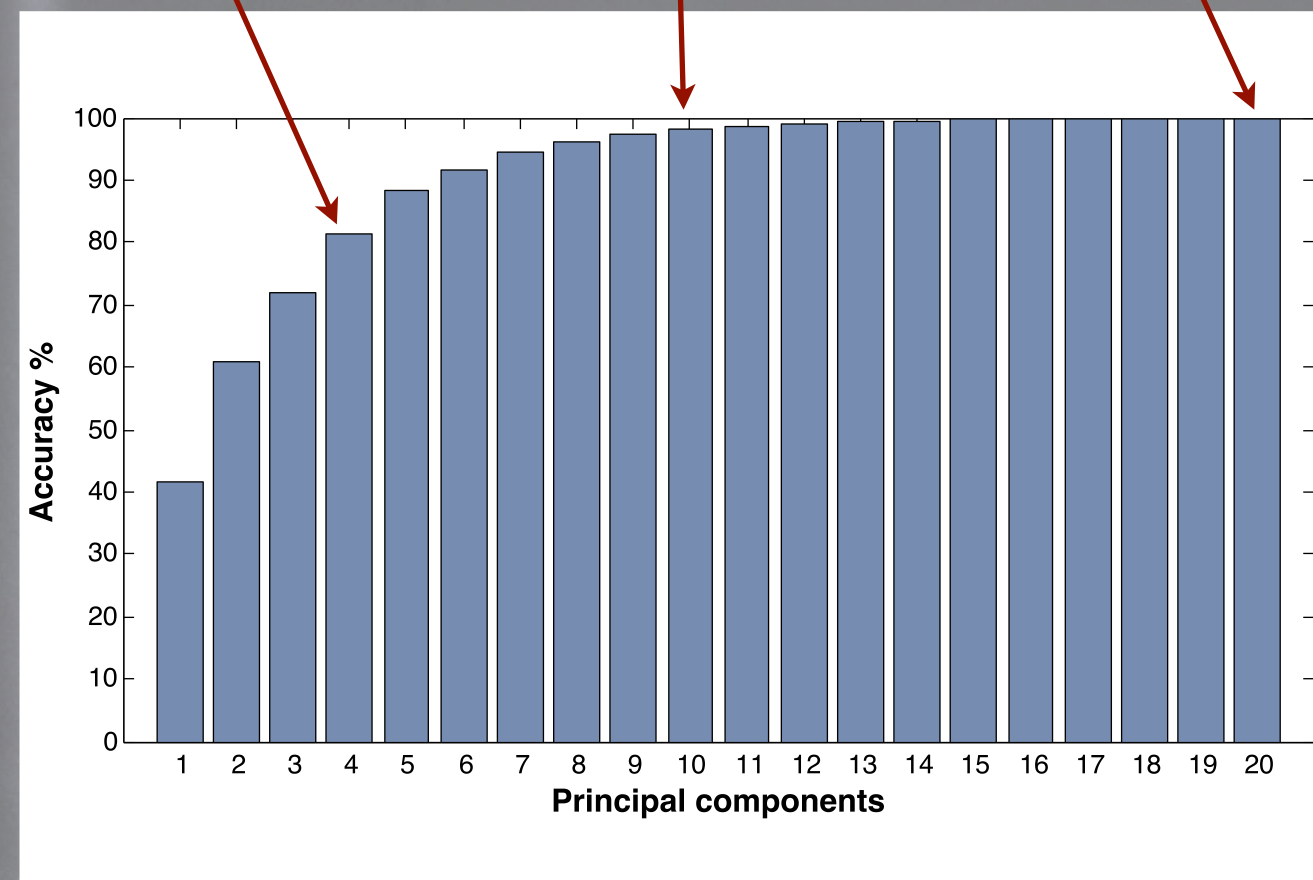


Accuracy of 3-D eigenposes



Sidestep 3-D eigenposes

81.38% 98% 100%



Accuracy accumulation along the principal axes

Hyperdimensional cylindrical transformation

For $f \in \mathbb{R}^n$ when $n > 3$

$$f(d_1, d_2, d_3, \ldots, d_n)$$

$$f(x, y, z_1, \ldots, z_{n-2})$$

3-D mapping

$$f(x, y, z) \rightarrow f(\varphi, r, h)$$

$$\varphi = \arctan(\tfrac{y}{x})$$

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

$$h = z$$

Suppose $f \in \mathbb{R}^5$

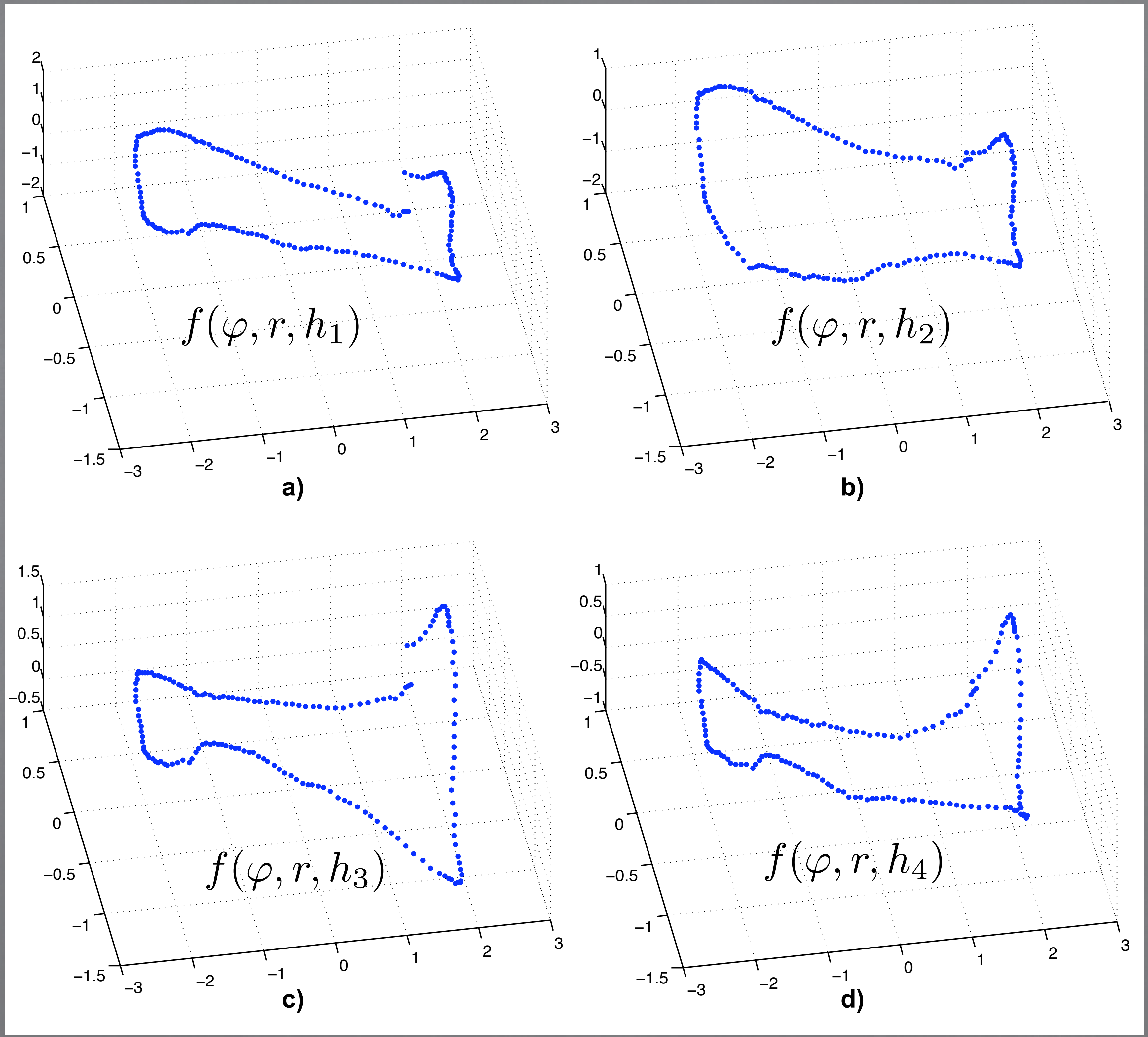
$$f(x, y, z_1, z_2, z_3)$$

$$\begin{matrix} f(x, y, z_1) & & f(\varphi, r, h_1) \\ f(x, y, z_2) & \Rightarrow & f(\varphi, r, h_2) \\ f(x, y, z_3) & & f(\varphi, r, h_3) \end{matrix}$$

Thus

$$f(x, y, z_1, \ldots, z_{n-2}) \rightarrow f(\varphi, r, h_1, \ldots, h_{n-2})$$

Multiple cylindrical frames



Hyperdimensional motion optimization

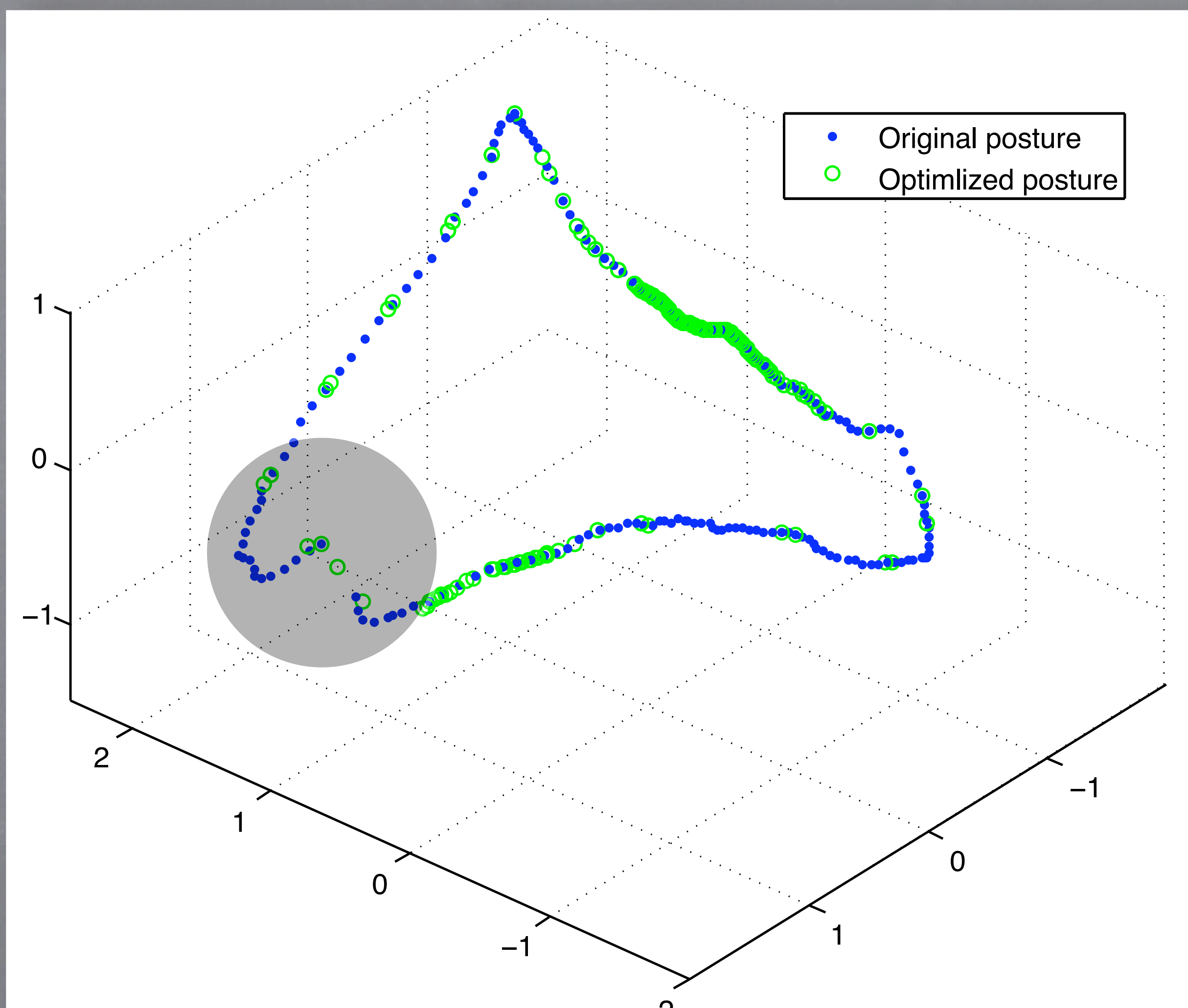
Hyperdimensional action subspace embedding

$$[r, h_1, h_2, \ldots, h_{18}] = g(\varphi)$$

Motion-phase optimization

$$\varphi_t^* = \arg\min_{\varphi_t} \Gamma(F(\omega_t, \omega_{t-1}, \varphi_t, \varphi_{t-1}))$$

Hyperdimensional optimization result



Conclusion

- Stable humanoid motion can be realized through imitation
- Compact low-dimensional spaces allows efficient optimization
- Dynamic model is not required

Conclusion

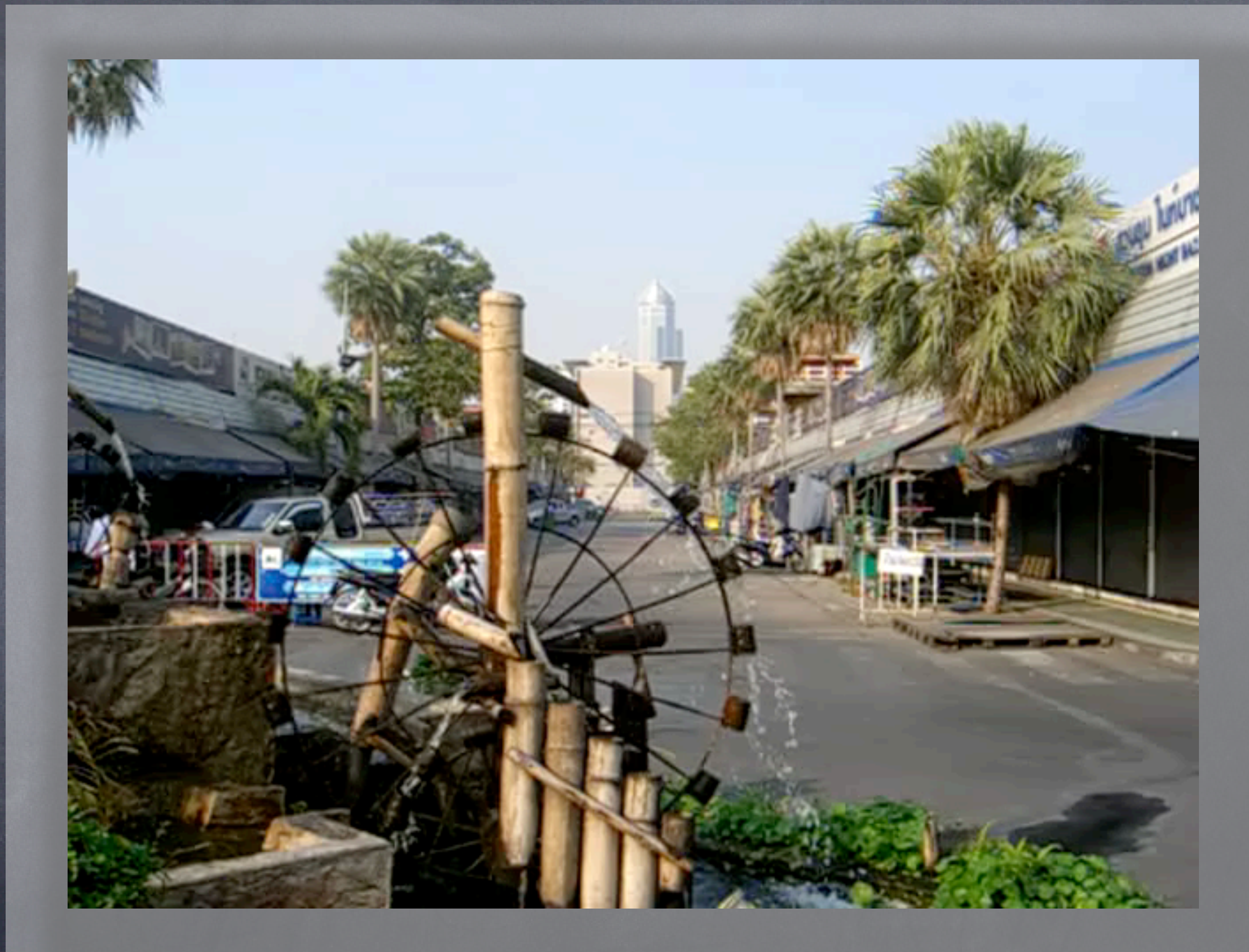
- Stable humanoid motion can be realized through imitation
- Compact low-dimensional spaces allows efficient optimization
- Dynamic model is not required

Note:

- Learn directly from the real robot
- Learn none-periodic motion
- Real-time feedback needs to be realized
- Multiple learning modules organization

Last but not least

Last but not least



Last but not least

Thank you!