Dynamic movement primitives

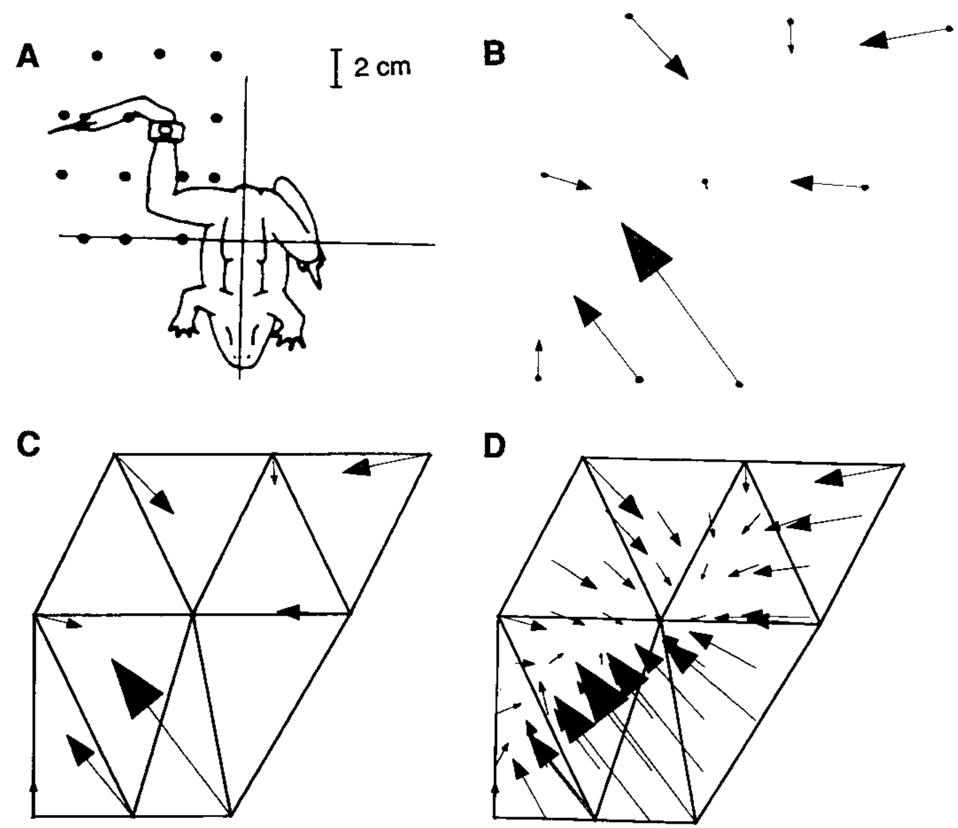
Gregor Schöner
gregor.schoener@ini.rub.de

Neural motivation

- Notion that neural networks in the brain and spinal cord generated a limited set of temporal templates
- whose weighted superposition is used to generate any given movement

Evidence for “primitives” in frog spinal cord

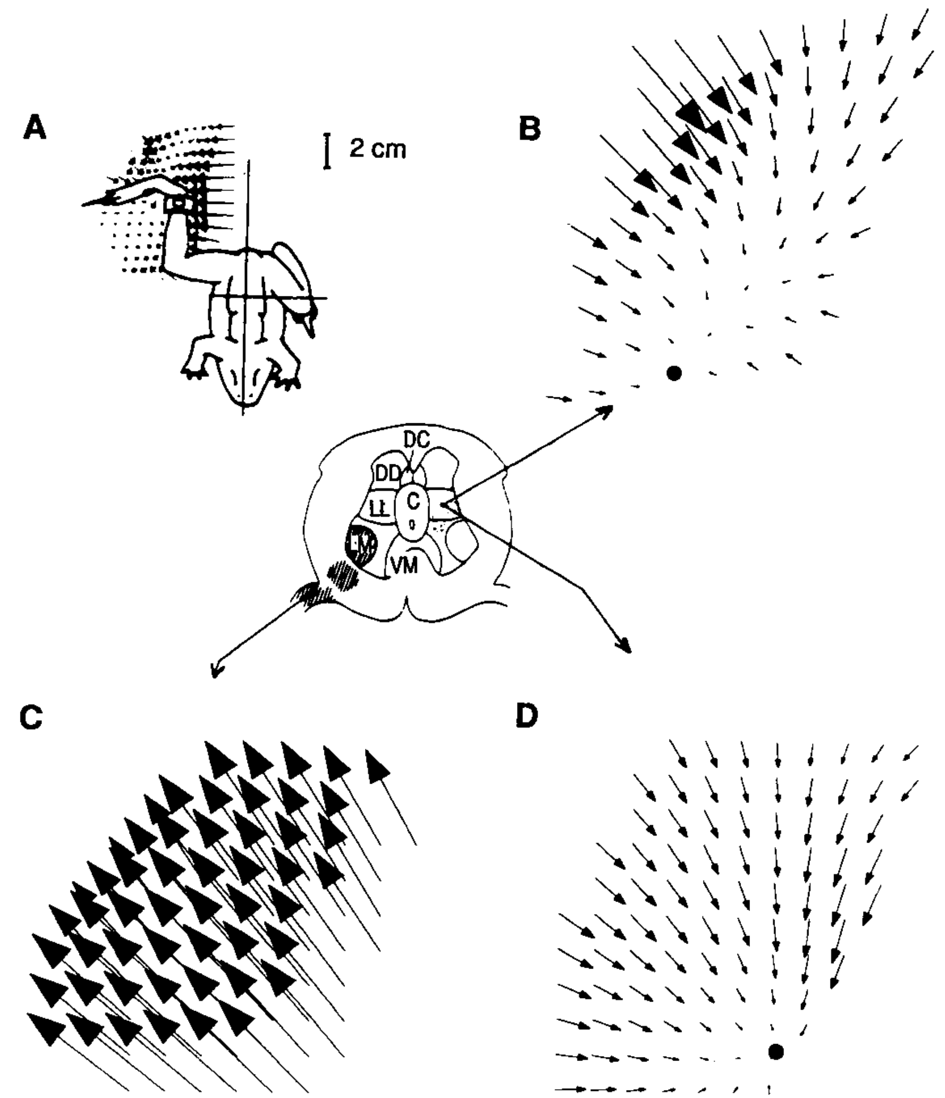
- electrical simulation in premotor spinal cord
- measure forces of resulted muscle activation pattern at different postures of limb
- interpolate force-field



[Bizzi, Mussa-Ivaldi, Gizter, 1991]

Evidence for "primitives" in frog spinal cord

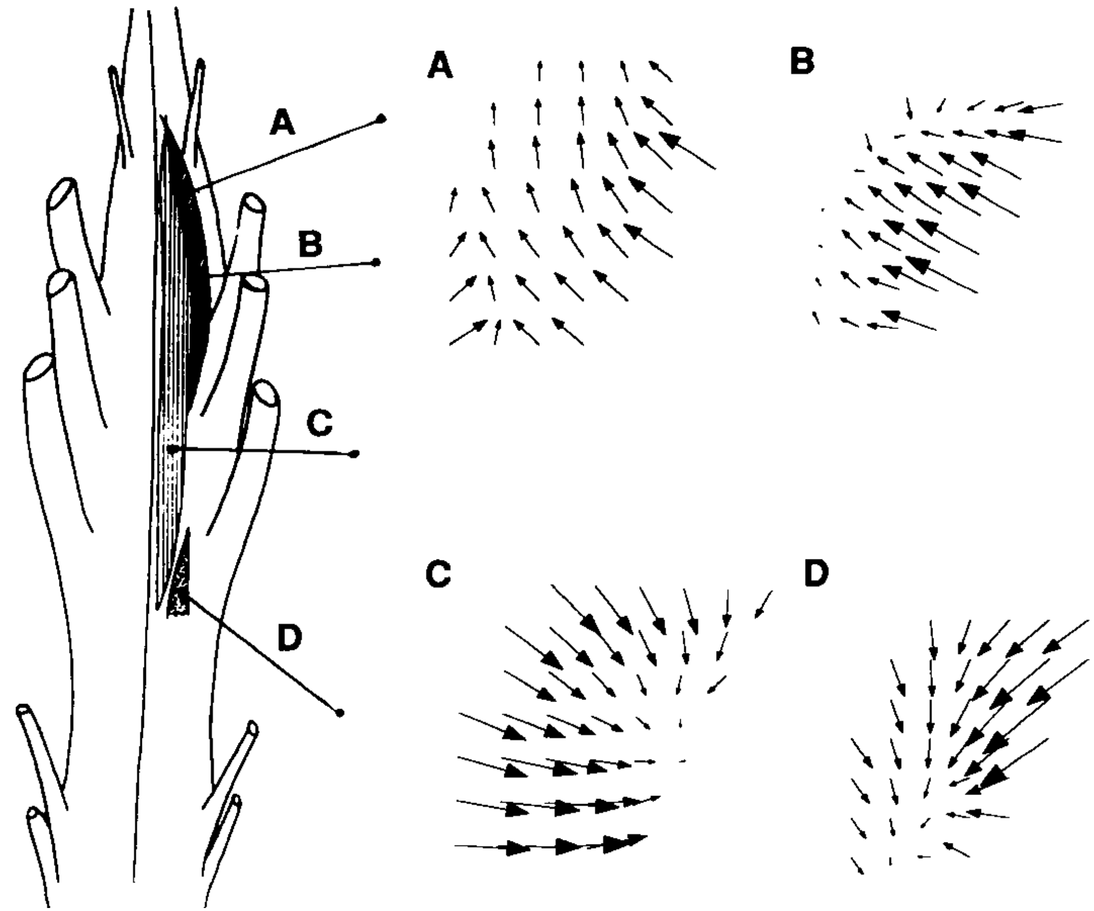
- parallel force-fields in premotor ares vs. convergent force fields from interneurons…



[Bizzi, Mussa-Ivaldi, Gizter, 1991]

Evidence for "primitives" in frog spinal cord

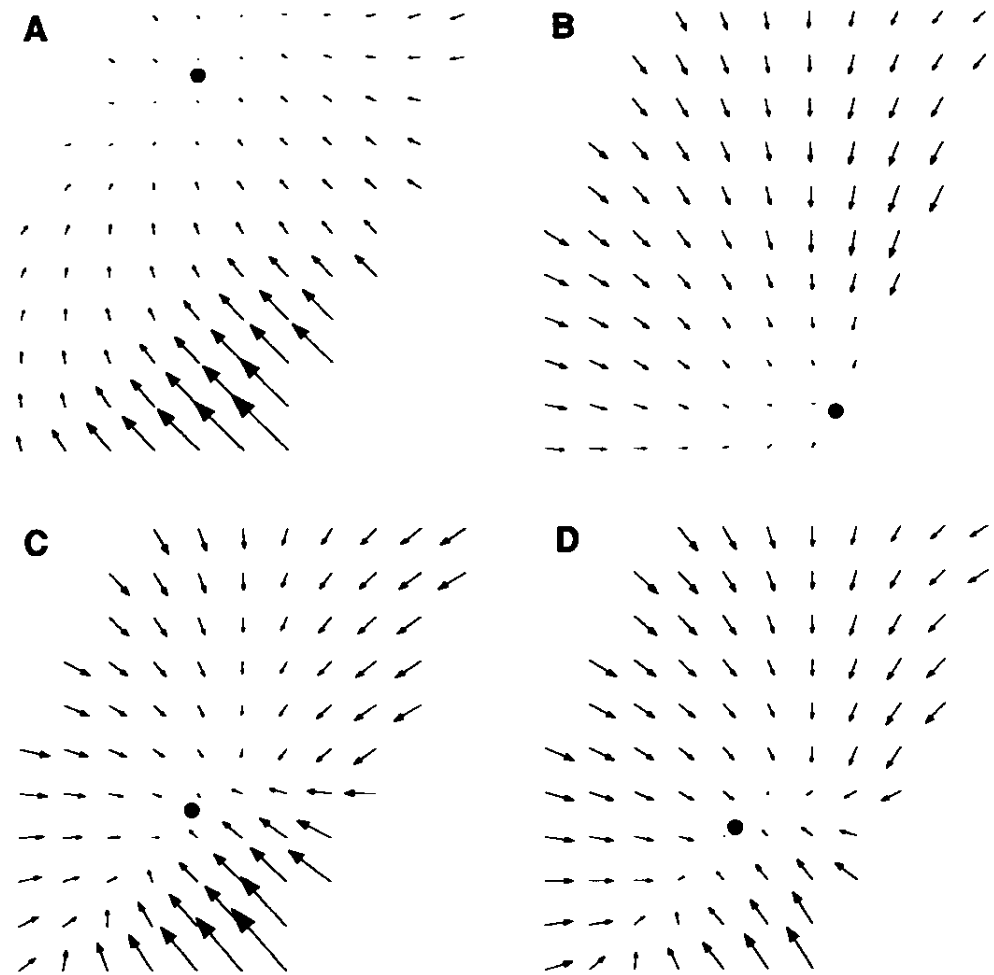
- convergent force-fields occur more often than expected by chance



[Bizzi, Mussa-Ivaldi, Gizter, 1991]

Evidence for "primitives" in frog spinal cord

- superposition of force-fields from joint stimulation



superposition of A and B

stimulating both A and B locations

[Bizzi, Mussa-Ivaldi, Gizter, 1991]

Dynamic movement primitives

[Ijspeert et al., Neural Computation 25:328-373 (2013)]

Base oscillator

- damped harmonic oscillator

$$\tau \ddot{y} = \alpha_z(\beta_z(g - y) - \dot{y}) + f,$$

y: position

- written as two first order equations

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f,$$

$$\tau \dot{y} = z,$$

z: velocity

- has fixed point attractor

$$(z, y) = (0, g)$$

g: goal point

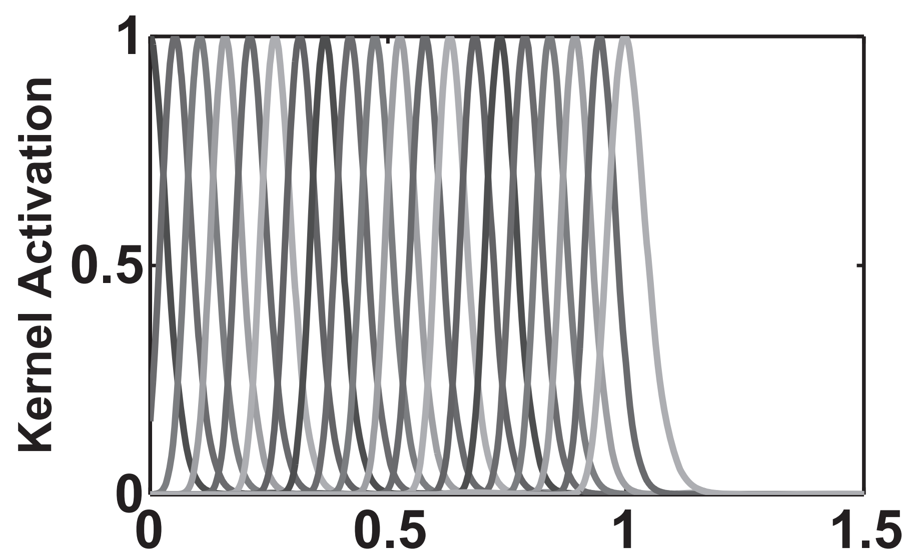
[Ijspeert et al., Neural Computation 25:328-373 (2013)]

Forcing function

- base functions
- weighted superposition makes forcing function
- which are explicit functions of time!
- => non-autonomous
- and, through c_i, also staggered in time, so there is a "score" being kept in time

$$\Psi_i(x) = \exp\left(-\frac{1}{2\sigma_i^2}(x - c_i)^2\right),$$

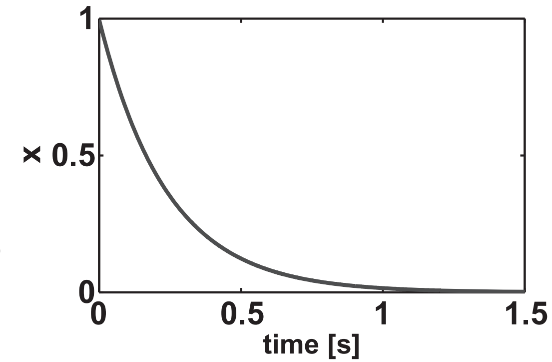
$$f(t) = \frac{\sum_{i=1}^{N} \Psi_i(t) w_i}{\sum_{i=1}^{N} \Psi_i(t)}$$



"Canonical system"

- "phase" variable, x, to (seemingly) get rid of non-autonomous character of dynamics

$$\tau \dot{x} = -\alpha_x x,$$



- but: … x is reset to an initial condition at each new movement initiation x(0)=1

$$f(x) = \frac{\sum_{i=1}^{N} \Psi_i(x) w_i}{\sum_{i=1}^{N} \Psi_i(x)} x(g - y_0)$$

- new: scale forcing functions with amplitude and with temporal distance from end of mov

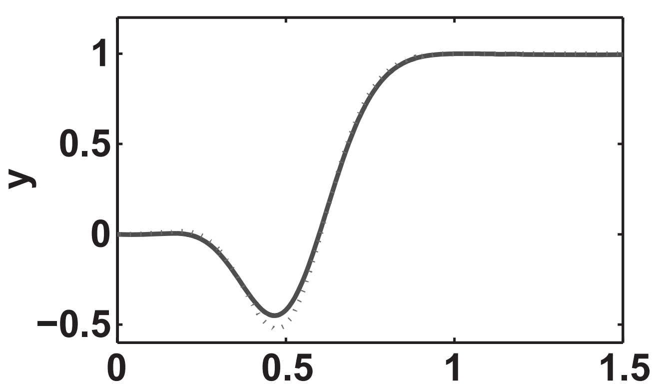
y_0 initial position

$g - y_0$ amplitude

Example 1D

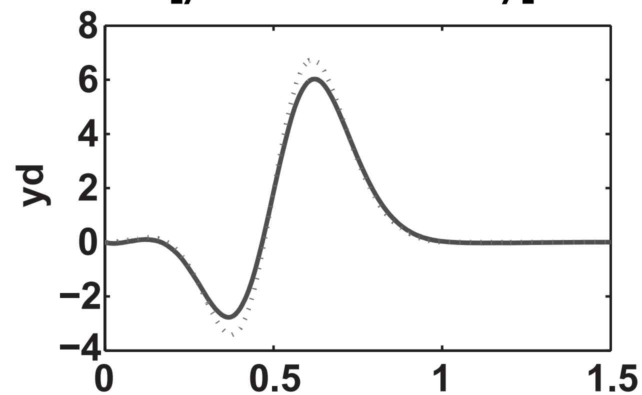
- weights fitted to track dotted trajectory (=5th order polynomial)… with first goes in the negative direction
- 20 kernels…

dotted: target
solid: approximation



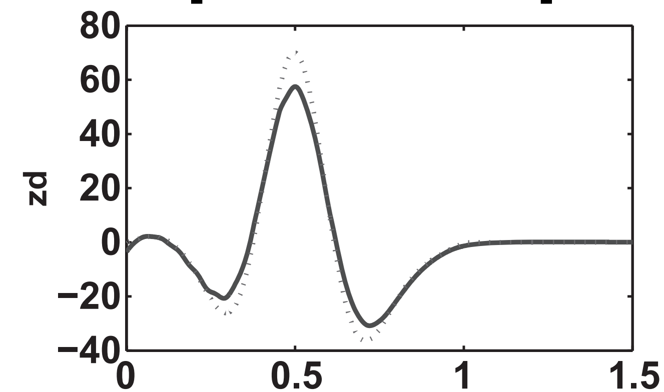
position

[yd: derivative of y]



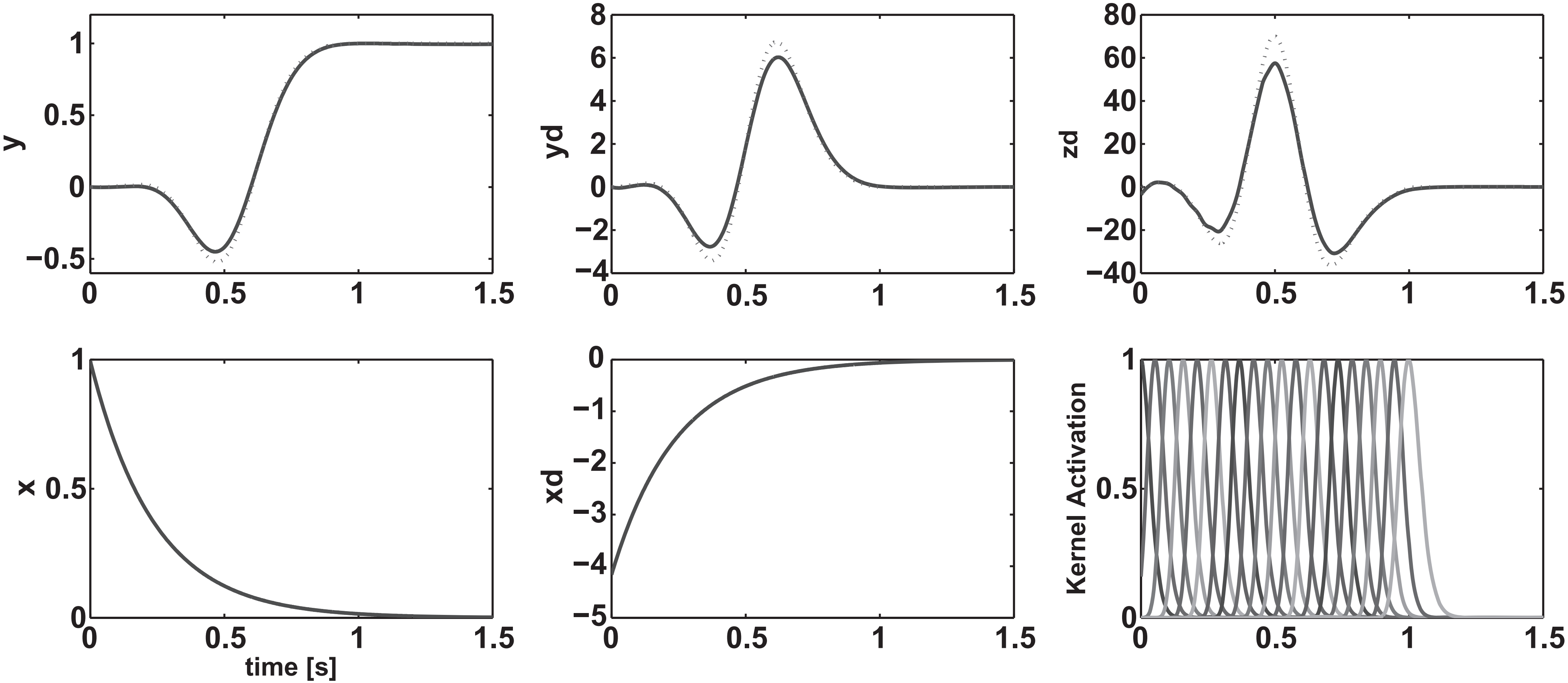
velocity

[zd: derivative of z]



acceleration

Example 1D



The space-time planning problem

- is to make sure the movement plan arrives at the target in a given time…
- the spatial goal is implemented by setting an attractor at the goal state
- the movement time is implicitly encoded in the tau/time scale of the "timing" variable…
  - but that relies on cutting off the timing variable, x, as some threshold level… as exponential time course never reaches zero…
  - quite sensitive to that threshold…

Periodic movement

- trivial phase oscillator (cycle time, tau) $\tau\dot{\phi} = 1,$
- trivial amplitude, r (constant), can be modulated by explicit time dependence
- forcing-function are functions of phase and amplitude

$$f(\phi, r) = \frac{\sum_{i=1}^{N} \Psi_i w_i}{\sum_{i=1}^{N} \Psi_i} r,$$

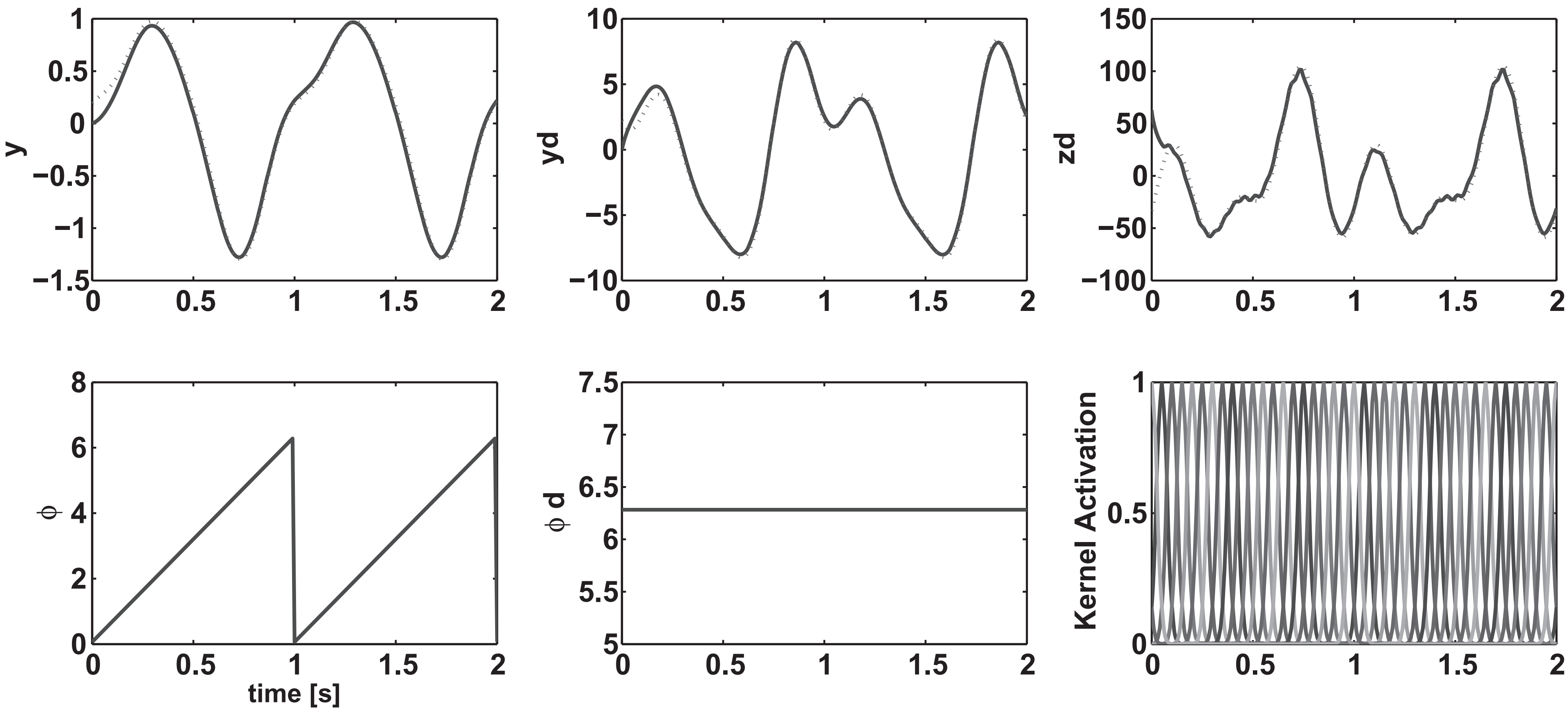
$$\Psi_i = \exp(h_i(\cos(\phi - c_i) - 1))$$

- base oscillator

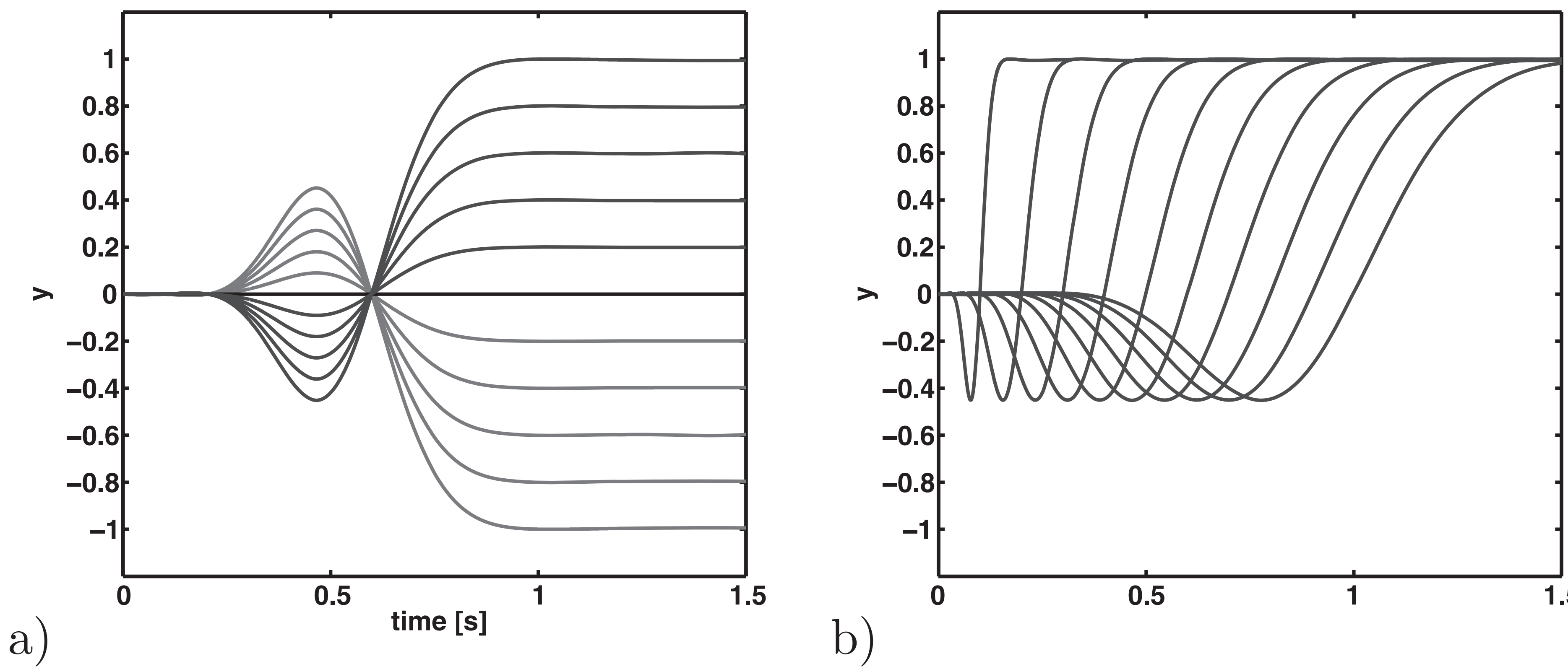
$$\tau\dot{z} = \alpha_z(\beta_z(g - y) - z) + f,$$

$$\tau\dot{y} = z,$$

Example: rhythmic movement



Scaling primitives

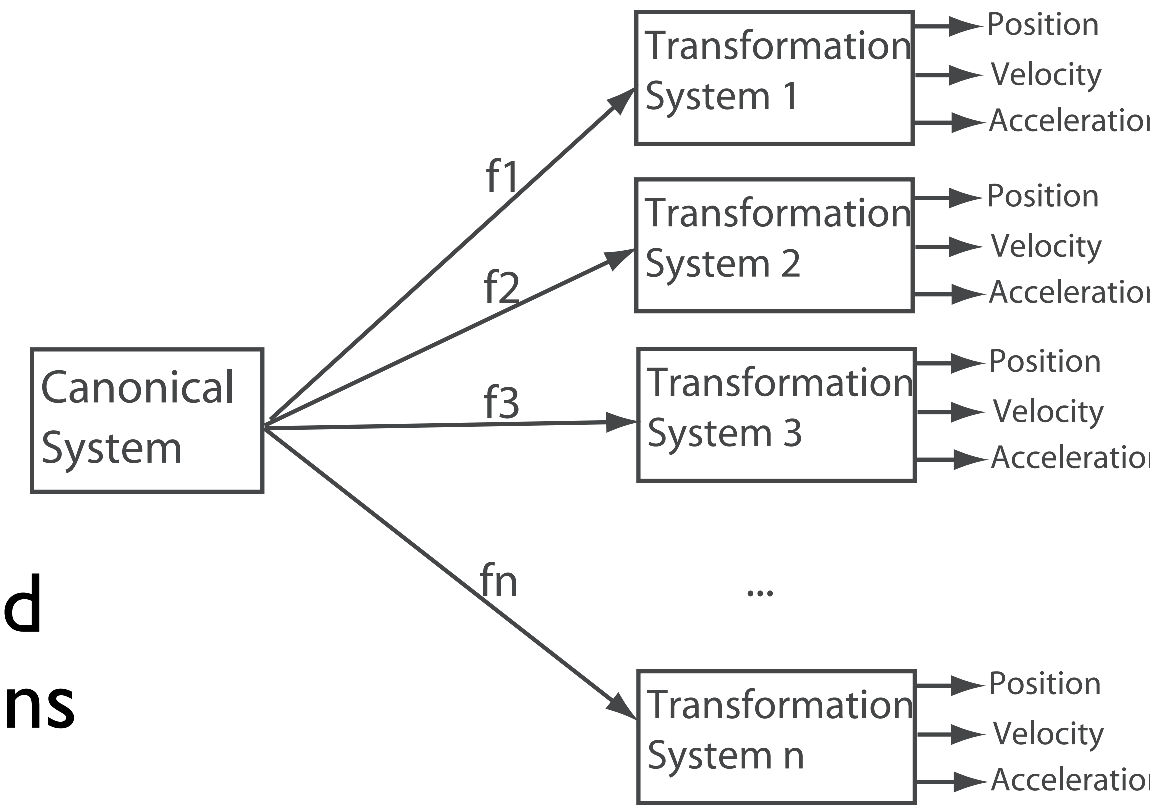


scale in space from -1 to 1

scale time from 0.15 to 1.7
but: not trivially right

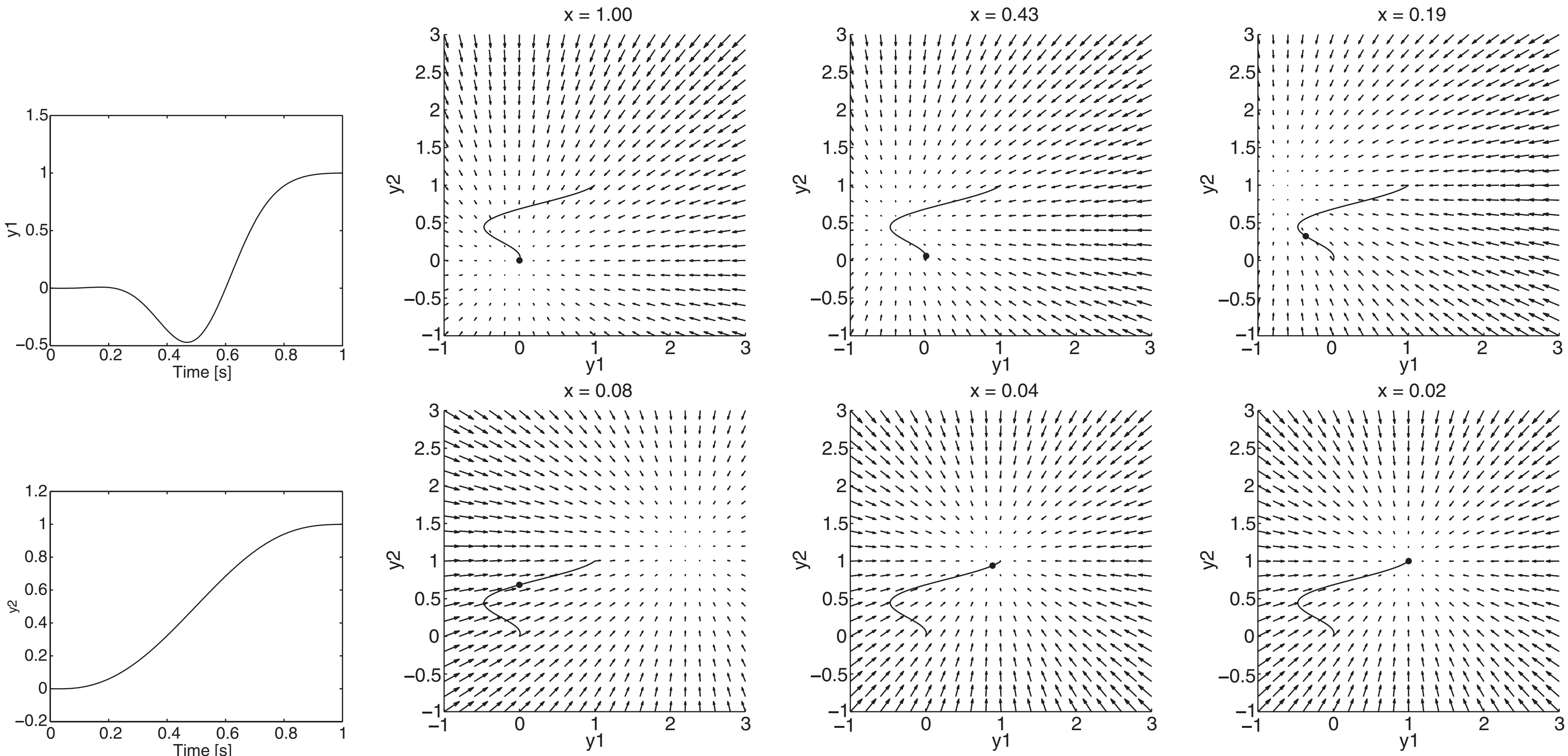
Multi-dimensional trajectories

- rather than couple multiple movement generator (deemed “complicated”)…
- only one central harmonic oscillator and multiple transformations of that…



Example 2D

- single “phase” x
- two base oscillator systems y1, y2
- with two sets of forcing functions



Learning the weights

$$[\ \tau \ddot{y} = \alpha_z(\beta_z(g - y) - \dot{y}) + f,\ \ ]$$

- base oscillator

$$f_{target} = \tau^2 \ddot{y}_{demo} - \alpha_z(\beta_z(g - y_{demo}) - \tau \dot{y}_{demo}).$$

- forcing function from sample trajectory

$$[\quad f(x) = \frac{\sum_{i=1}^{N} \Psi_i(x) w_i}{\sum_{i=1}^{N} \Psi_i(x)} x(g - y_0) \quad]$$

- weights by minimizing error J

$$J_i = \sum_{t=1}^{P} \Psi_i(t)(f_{target}(t) - w_i \xi(t))^2,$$

$$\xi(t) = x(t)(g - y_0)$$ for discrete mov

$$\xi(t) = r$$ for rhythmic mov

Learning the weights

- can be solved analytically

minimum of

$$J_i = \sum_{t=1}^{P} \Psi_i(t)(f_{target}(t) - w_i \xi(t))^2,$$

$$\xi(t) = x(t)(g - y_0)$$

$$\xi(t) = r$$

is

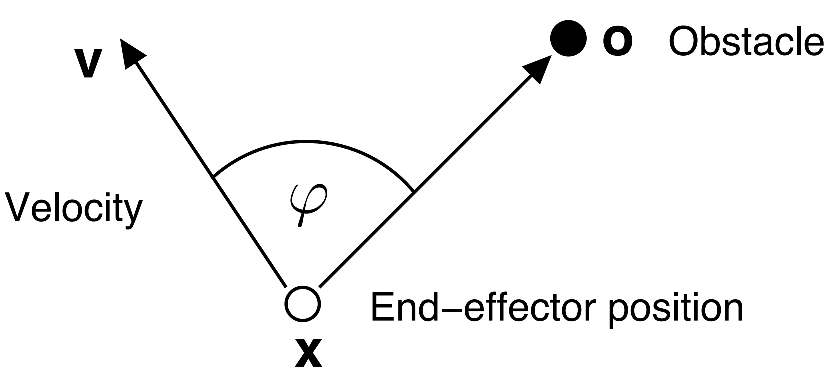
$$w_i = \frac{\mathbf{s}^T \mathbf{\Gamma}_i \mathbf{f}_{target}}{\mathbf{s}^T \mathbf{\Gamma}_i \mathbf{s}},$$

where (P=# sample times in demo trajectories):

$$\mathbf{s} = \begin{pmatrix} \xi(1) \\ \xi(2) \\ \dots \\ \xi(P) \end{pmatrix} \quad \mathbf{\Gamma}_i = \begin{pmatrix} \Psi_i(1) & & & 0 \\ & \Psi_i(2) & & \\ & & \dots & \\ 0 & & & \Psi_i(P) \end{pmatrix} \quad \mathbf{f}_{target} = \begin{pmatrix} f_{target}(1) \\ f_{target}(2) \\ \dots \\ f_{target}(P) \end{pmatrix}$$

Obstacle avoidance

- inspired by Schöner/ Dose (in Fajen Warren form)
- obstacle avoidance force-let



$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f + C_t,$$

$$\tau \dot{y} = z.$$

$$\mathbf{C}_t = \gamma \mathbf{R}\dot{\mathbf{y}}\,\theta \exp(-\beta\theta),$$

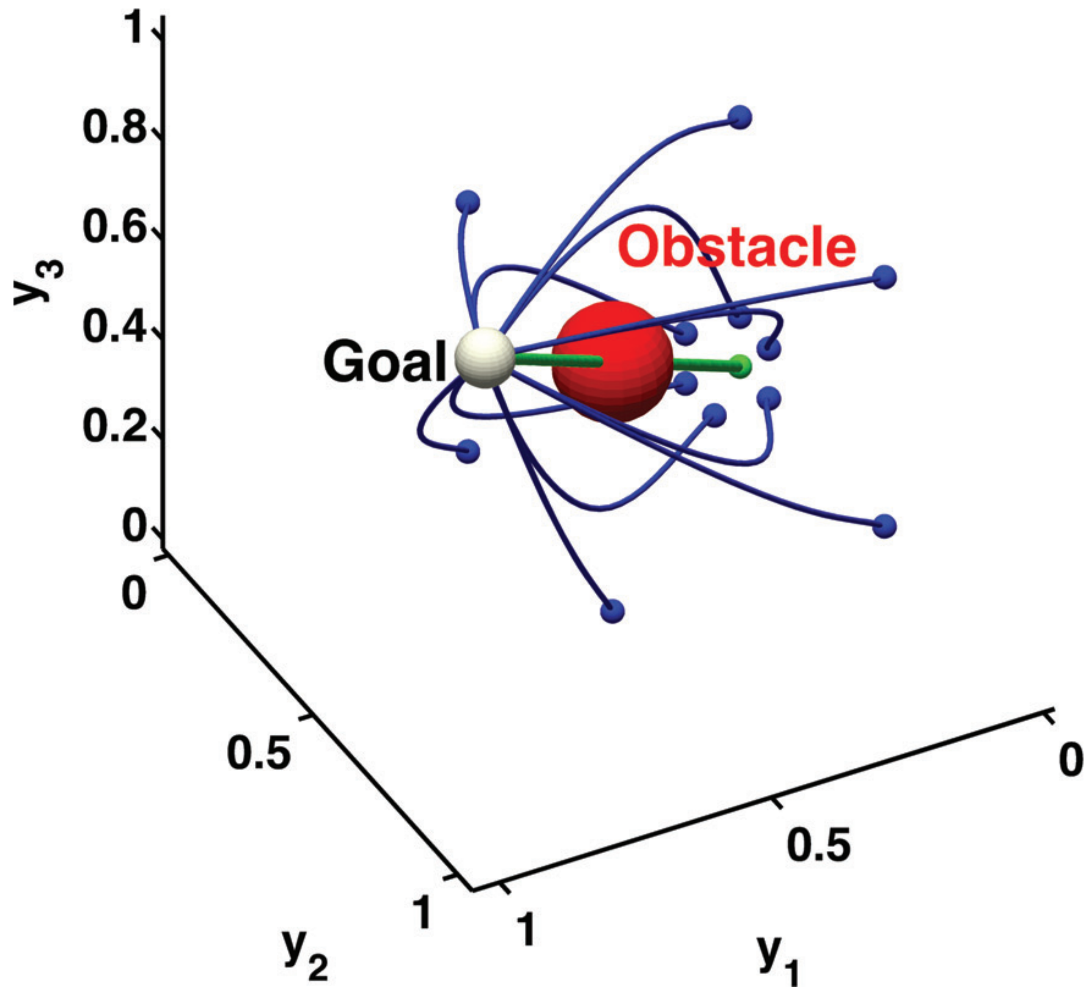
where

$$\theta = \arccos\left(\frac{(\mathbf{o} - \mathbf{y})^T \dot{\mathbf{y}}}{|\mathbf{o} - \mathbf{y}||\dot{\mathbf{y}}|}\right),$$

$$\mathbf{r} = (\mathbf{o} - \mathbf{y}) \times \dot{\mathbf{y}}.$$

[actually this is: Reimann, Iossifidis, Schöner, 2010]

Obstacle avoidance



But: human obstacle avoidance is not really like that…

- => Grimme, Lipinski, Schöner, 2012

Coordination

- in phase dynamics: couple to external timers…
- but: issue of predicting such events and aligning the prediction to achieve synchronicity…

$$\tau \dot{x} = -\alpha_x x + C_c$$

$$\tau \dot{\phi} = 1 + C_c.$$

$$C_c = \alpha_c (\phi_{ext} - \phi).$$

Conclusion

- DMP enable learning "movement styles" while enabling generalization to new movement targets
- DMP is a purely kinematic account
- => DMP is not addressing control
  - in that respect, analogy to force-fields is misleading
- DMP addresses timing, but account of coordination is limited
- DMP for different tasks and their combination… open