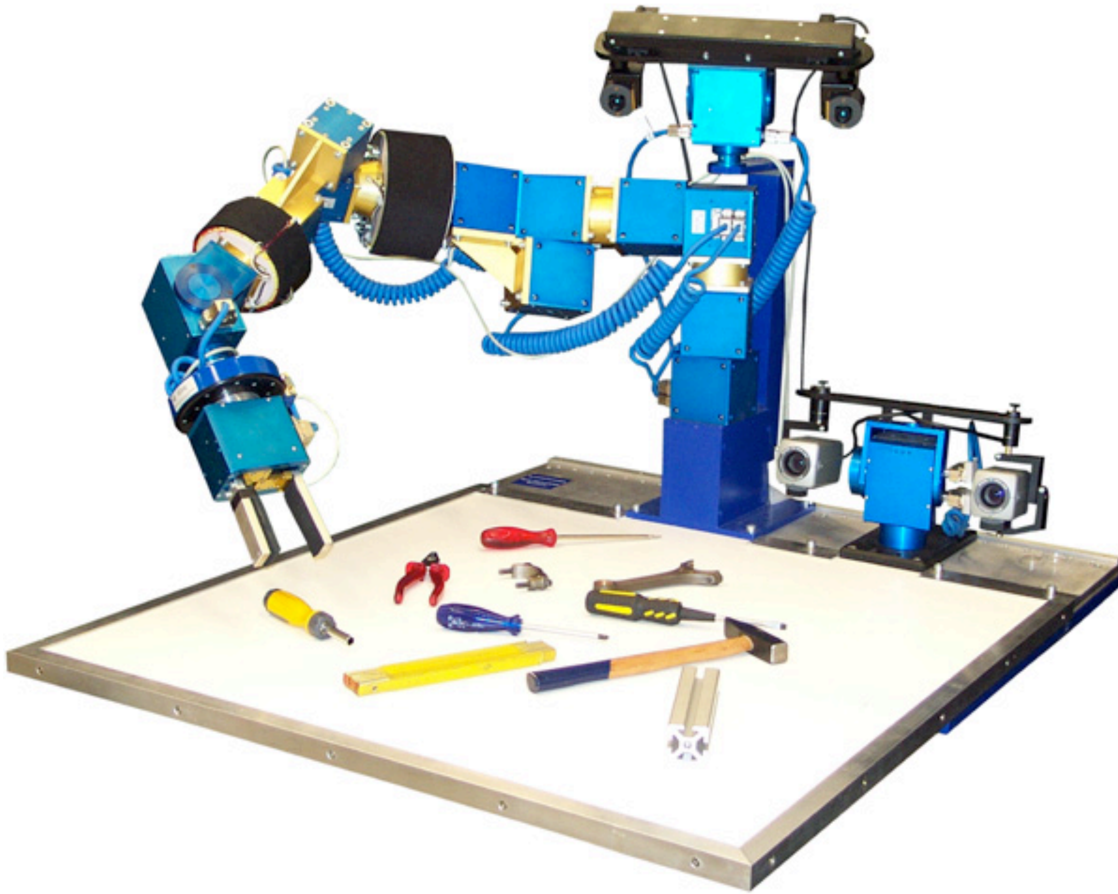
The degree of freedom problem

Gregor Schöner
gregor.schoener@ini.rub.de

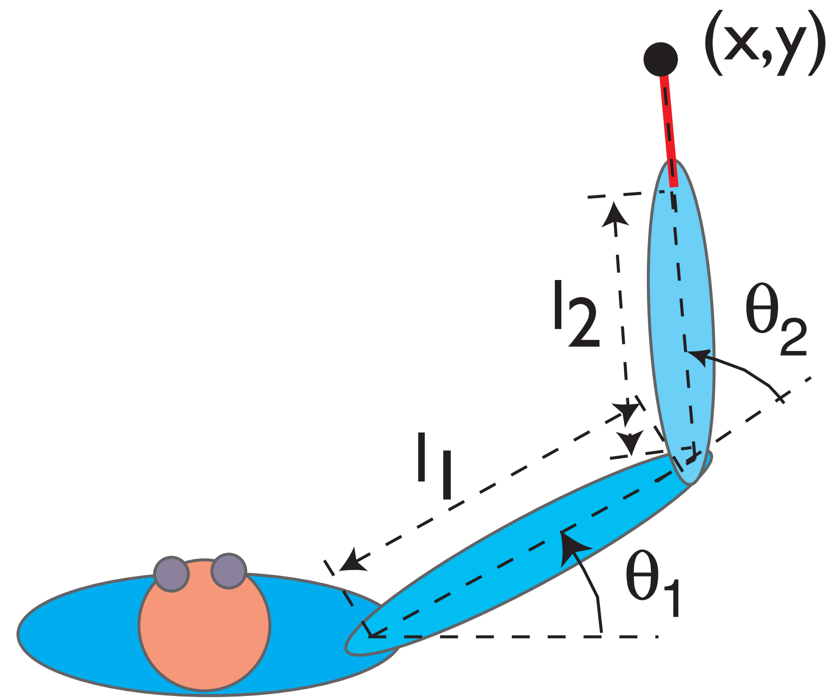
Spaces for robotic motion planning

- task level planning is about end-effector pose in space (e.g,. 3 translational and 3 rotational degrees of freedom)
- configuration space planning: joint angles of actuated degrees of freedom



Forward kinematics

- where is the hand, given the joint angles..



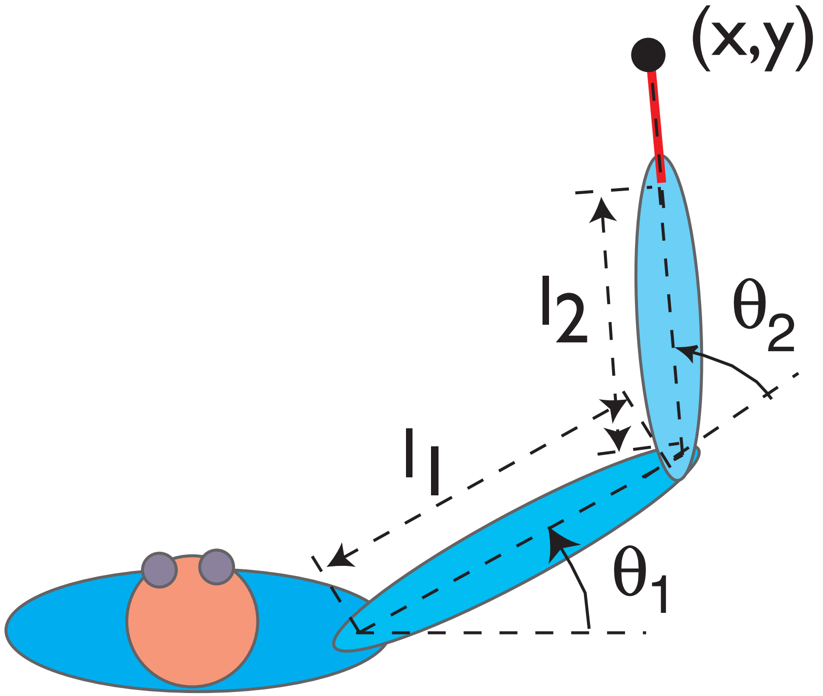
$$\mathbf{x} = \mathbf{f}(\theta)$$

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Differential forward kinematics

- where is the hand moving, given the joint angles and velocities



$$\dot{\mathbf{x}} = \mathbf{J}(\theta)\dot{\theta}$$

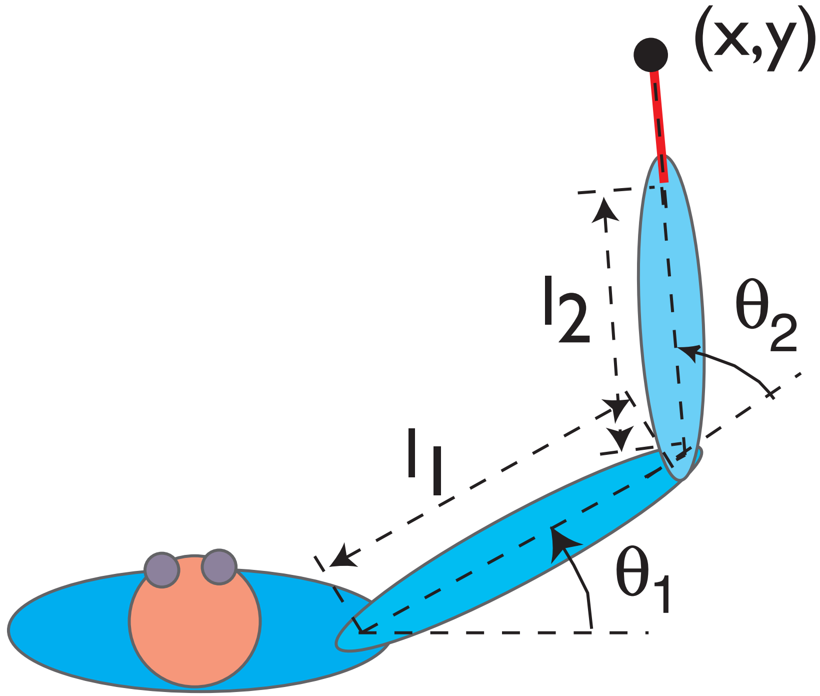
$$\dot{x} = -\, l_1 \sin(\theta_1)\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_1 - l_2 \sin(\theta_1 + \theta_2)\dot{\theta}_2$$

$$\dot{y} = l_1 \cos(\theta_1)\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)\dot{\theta}_2$$

Differential forward kinematics

- where is the hand moving, given the joint angles and velocities

$$\dot{\mathbf{x}} = \mathbf{J}(\theta)\dot{\theta}$$

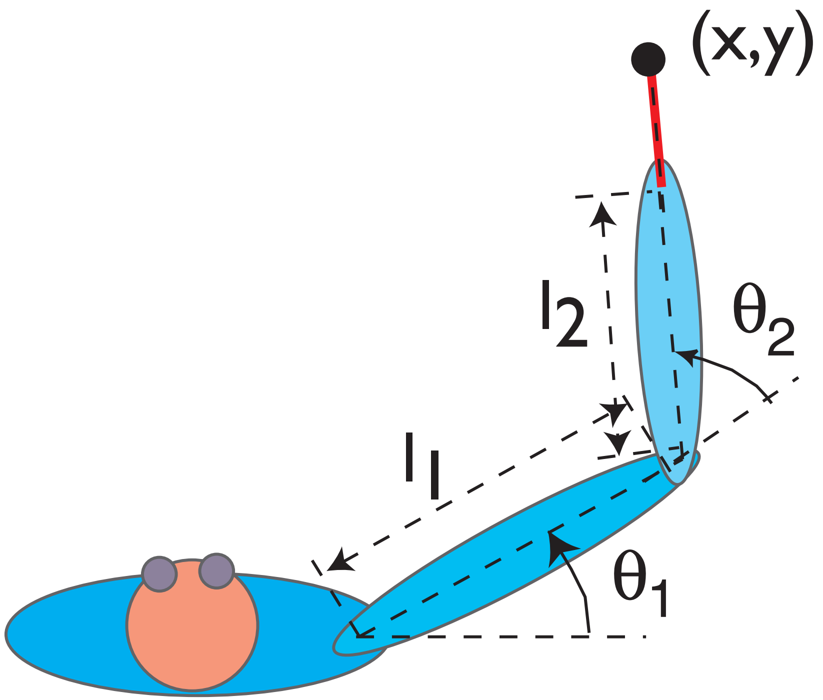


$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -l_1\cos(\theta_1) - l_1\cos(\theta_1+\theta_2) & -l_2\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \end{pmatrix} \cdot \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

Inverse kinematics

- what joint angles are needed to put the hand at a given location
- exact solution:

$$\theta = \mathbf{f}^{-1}(\mathbf{x})$$



Inverse kinematics

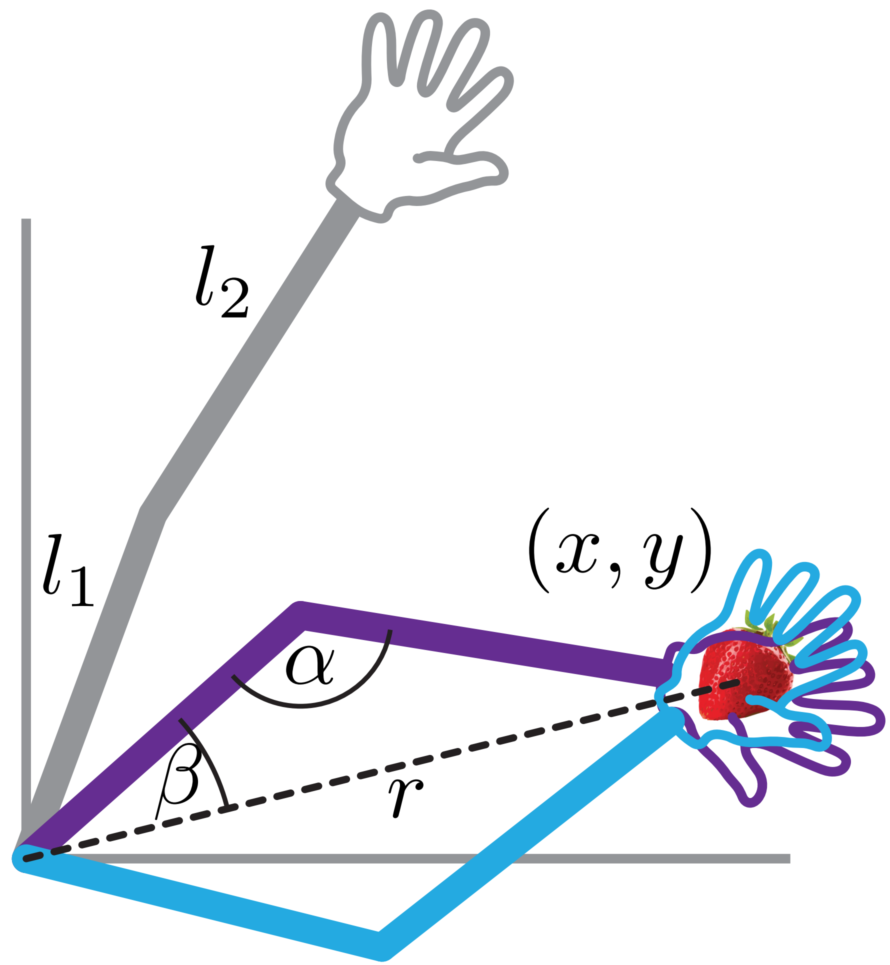
$$\theta_1 = \arctan_2(y, x) \pm \beta$$

$$\theta_2 = \pi \pm \alpha$$

$$\alpha = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - r^2}{2 l_1 l_2}\right)$$

$$\beta = \cos^{-1}\left(\frac{r^2 + l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

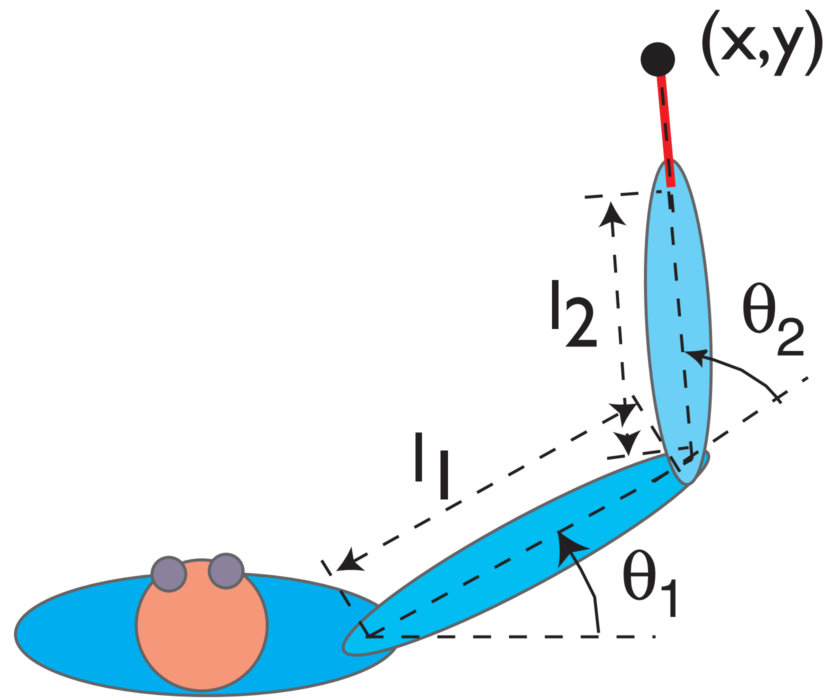
where $\quad r^2 = x^2 + y^2$



[thanks to Jean-Stéphane Jokeit]

Differential inverse kinematics

- which joint velocities to move the hand in a particular way



$$\dot{\theta} = \mathbf{J}^{-1}(\theta)\dot{\mathbf{x}}$$

with the inverse of

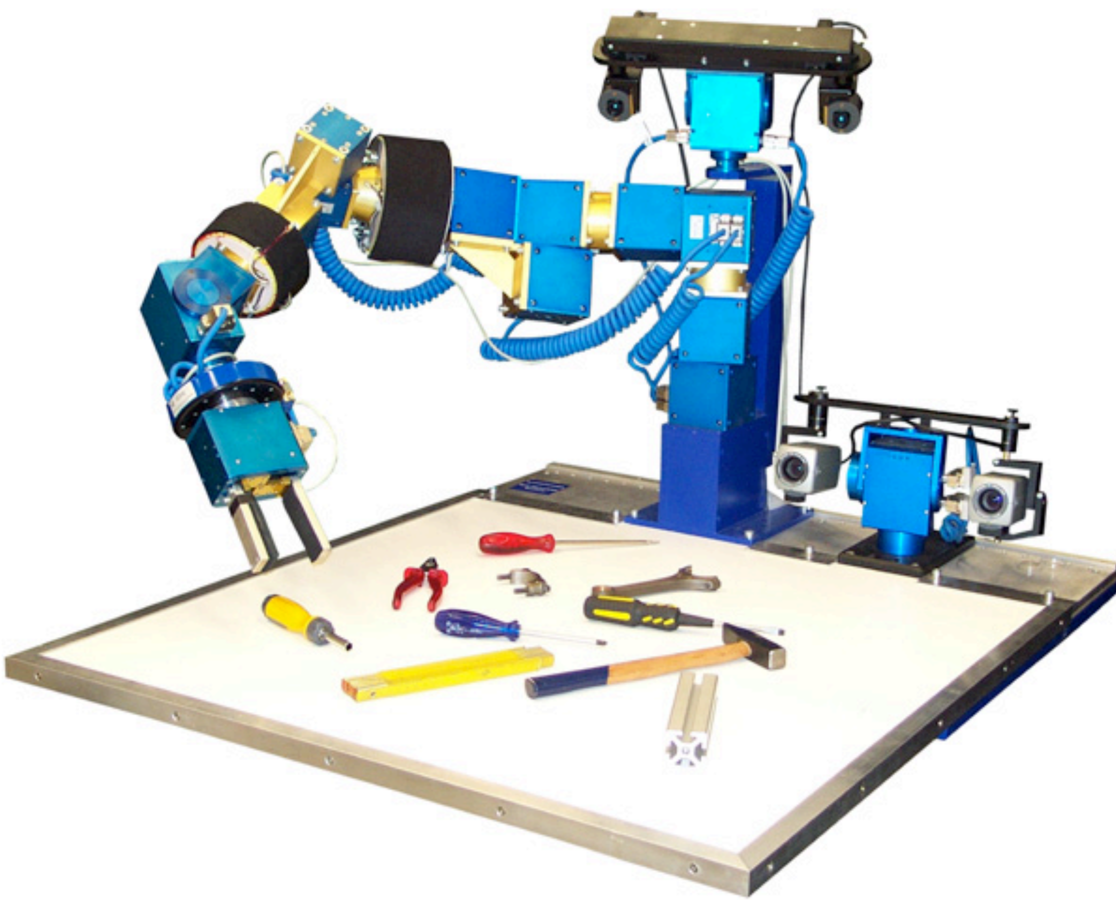
$$\mathbf{J}(\theta) = \begin{pmatrix} -l_1\cos(\theta_1) - l_1\cos(\theta_1+\theta_2) & -l_2\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \end{pmatrix}$$

if it exists!

Spaces for robotic motion planning

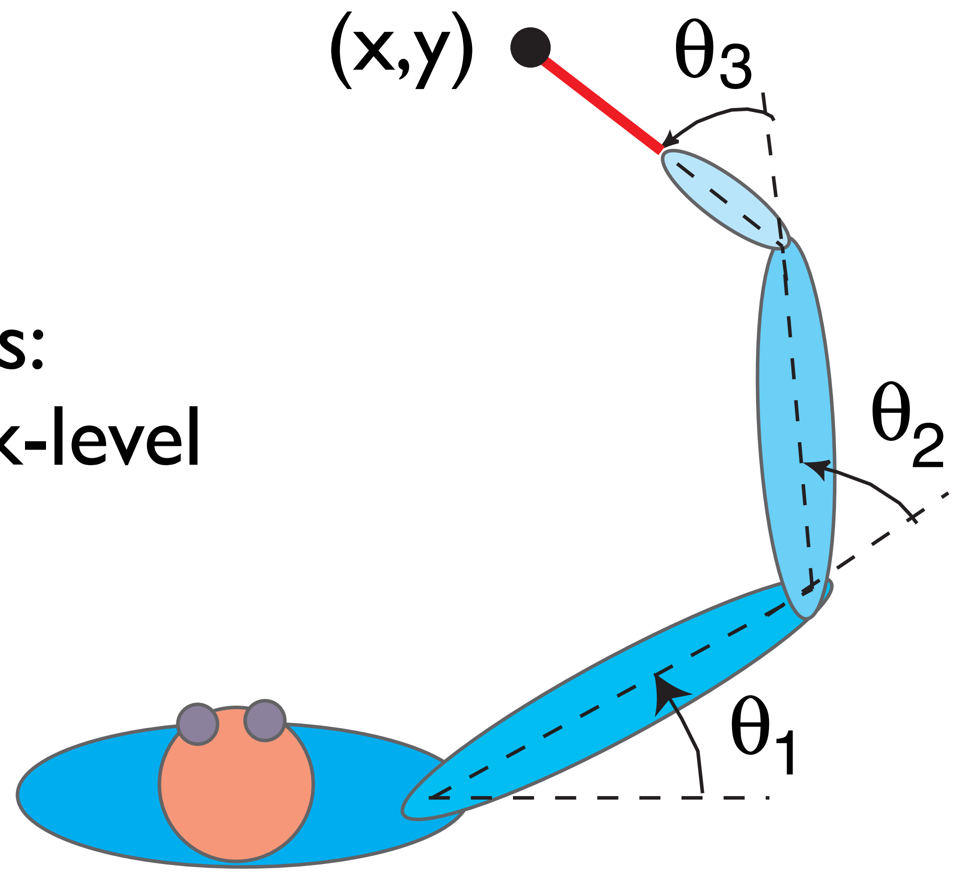
kinematic model	$\mathbf{x} = \mathbf{f}(\theta)$	$\dot{\mathbf{x}} = \mathbf{J}(\theta)\dot{\theta}$
inverse kinematic model	$\theta = \mathbf{f}^{-1}(\mathbf{x})$	$\dot{\theta} = \mathbf{J}^{-1}(\theta)\dot{\mathbf{x}}$

- transform end-effector to configuration space through inverse kinematics
- problems of singularities and multiple "leafs" of inverse…



Spaces for robotic motion planning

- redundant arms/tasks: more joints than task-level degrees of freedom

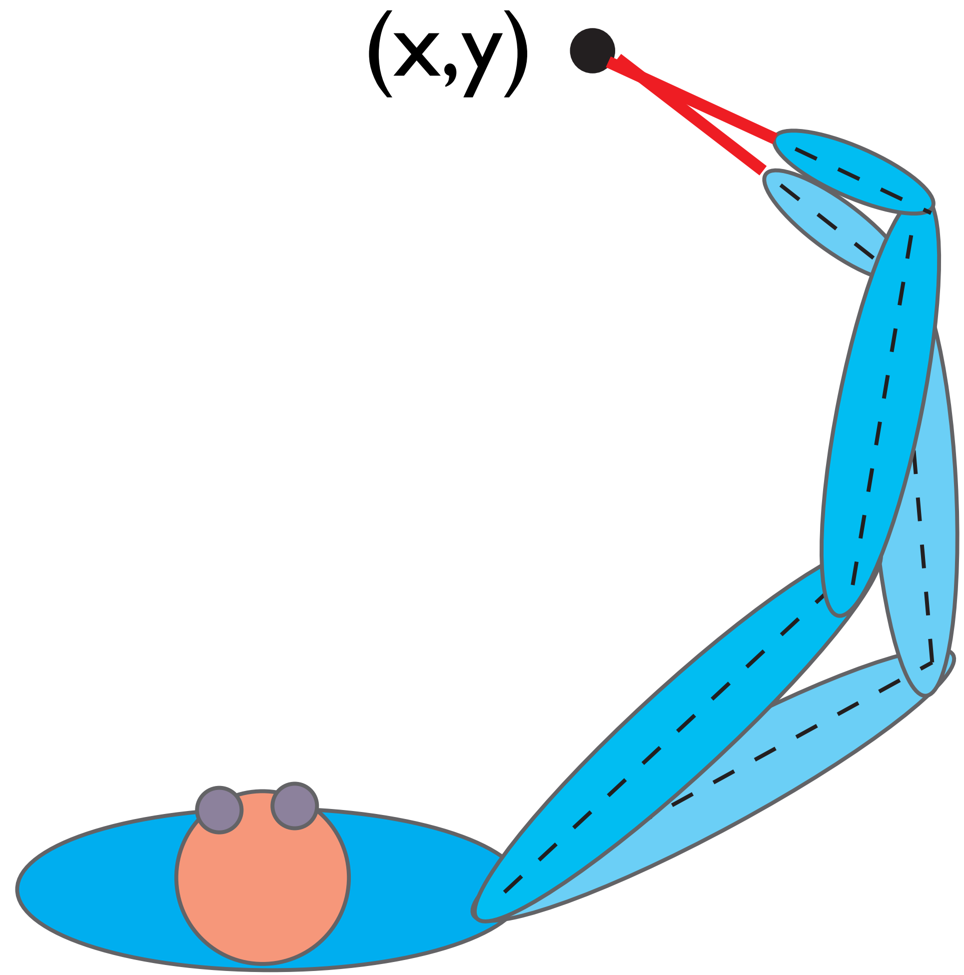


$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1+\theta_2) + l_3 \cos(\theta_1+\theta_2+\theta_3)$$

$$y = l_2 \sin(\theta_1) + l_2 \sin(\theta_1+\theta_2) + l_3 \sin(\theta_1+\theta_2+\theta_3)$$

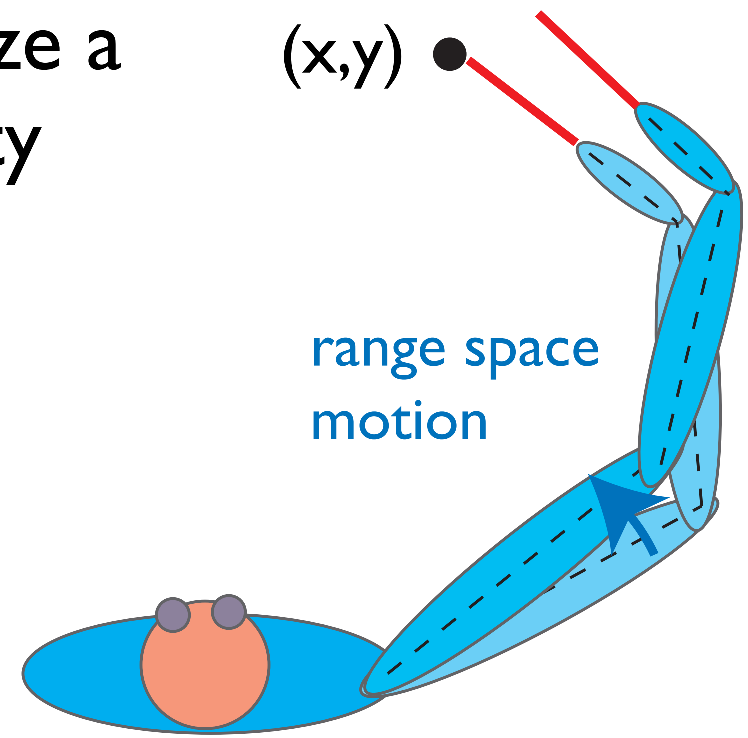
Spaces for robotic motion planning

- => (continuously) many inverse solutions…



Spaces for robotic motion planning

- use pseudo-inverses that minimize a functional (e.g., total joint velocity or total momentum)



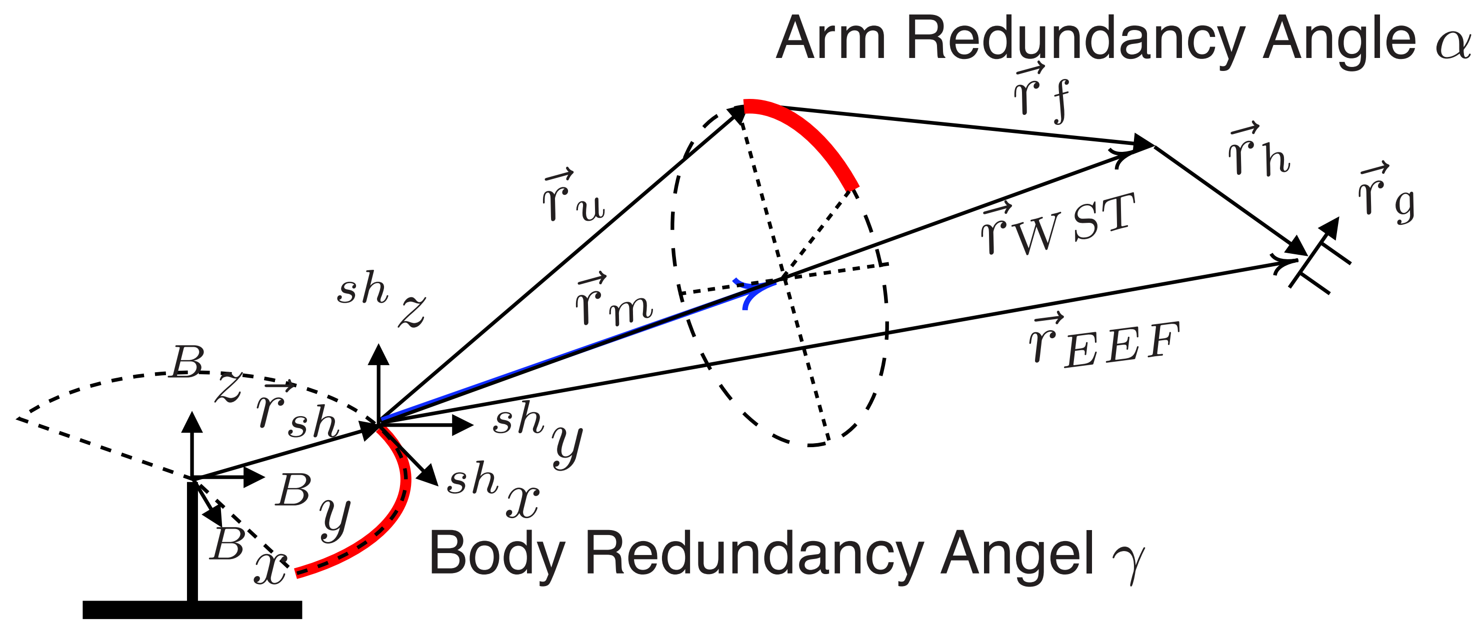
$$\dot{\mathbf{x}} = \mathbf{J}(\theta)\dot{\theta}$$

$$\dot{\theta} = \mathbf{J}^{+}(\theta)\dot{\mathbf{x}}$$

$$\mathbf{J}^{+}(\theta) = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}$$ pseudo-inverse

Spaces for robotic motion planning

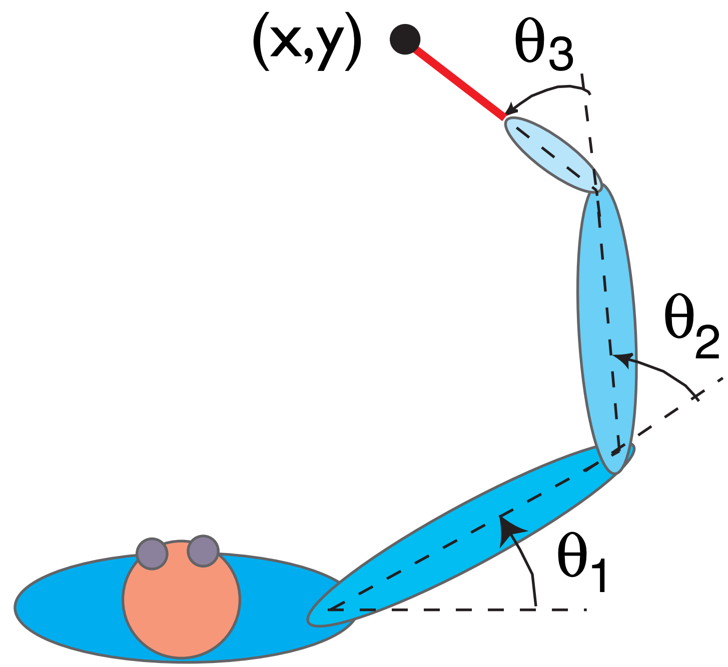
- or use extra degrees of freedom for additional tasks



[Iossifidis, Schöner, ICRA 2004]

Degree of freedom problem in human movement

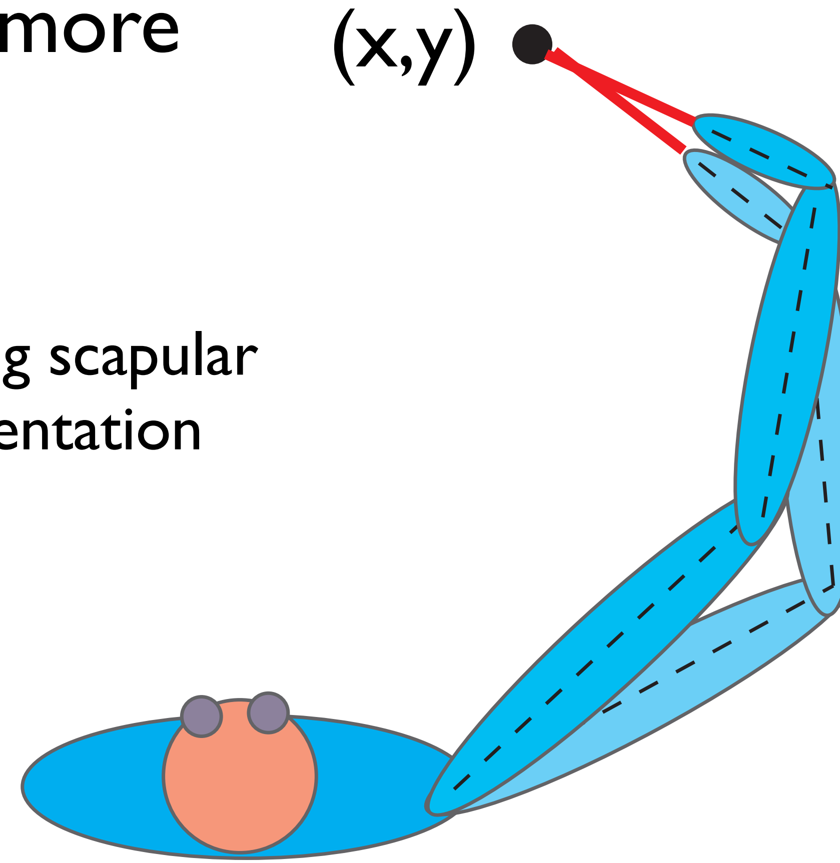
- what is a DoF?
  - variable that can be independently varied
  - e.g. joint angles
- muscles/muscle groups
  - but: assess to which extent they can be activated independently…
  - .. mode picture



$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1+\theta_2) + l_3 \cos(\theta_1+\theta_2+\theta_3)$$
$$y = l_2 \sin(\theta_1\ ) + l_2 \sin(\theta_1+\theta_2) + l_3 \sin(\theta_1+\theta_2+\theta_3)$$

Degree of freedom problem in human movement

- for most tasks, there are many more degrees of freedom than task constraints…
  - e.g., 10 joints in the upper arm including scapular joints to control hand position and orientation (3 to 5 or 6 DoF)
  - but typically more: involve upper trunk movements
  - or even make a step to move
- many muscles per joint (e.g. about 750 muscles in the human body vs. about 50 DoF)

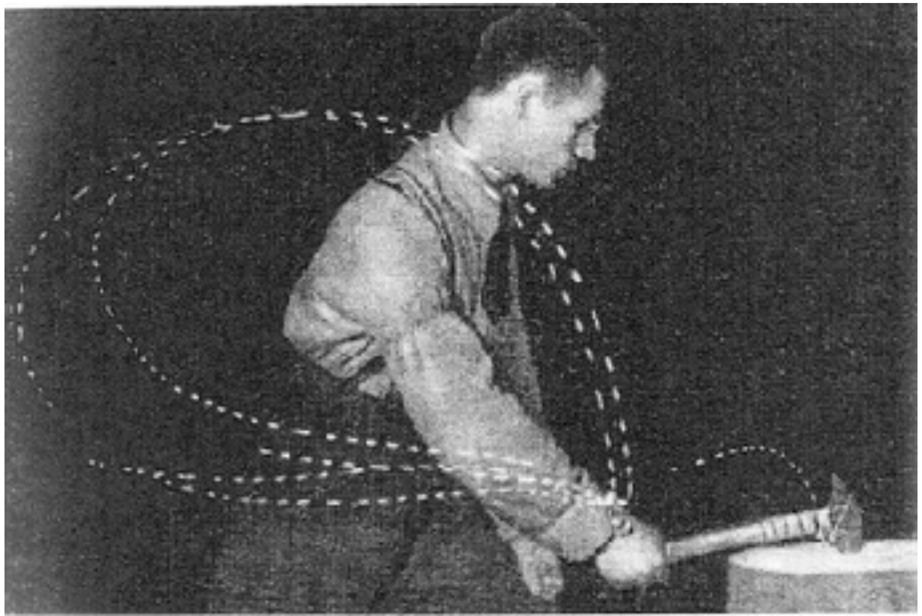


Degree of freedom problem in human movement

- Nikolai Bernstein… 1930's… in the Soviet Union
- "how to harness the many DoF to achieve the task"

Bernstein's workers

- highly skilled workers wielding a hammer to hit a nail… => hammer trajectory in space less variable than body configuration
  - as detected in superposing spatial trajectories of lights on hammer vs. on body..
  - but: camera frame anchored to nail/space, while initial body configuration varied

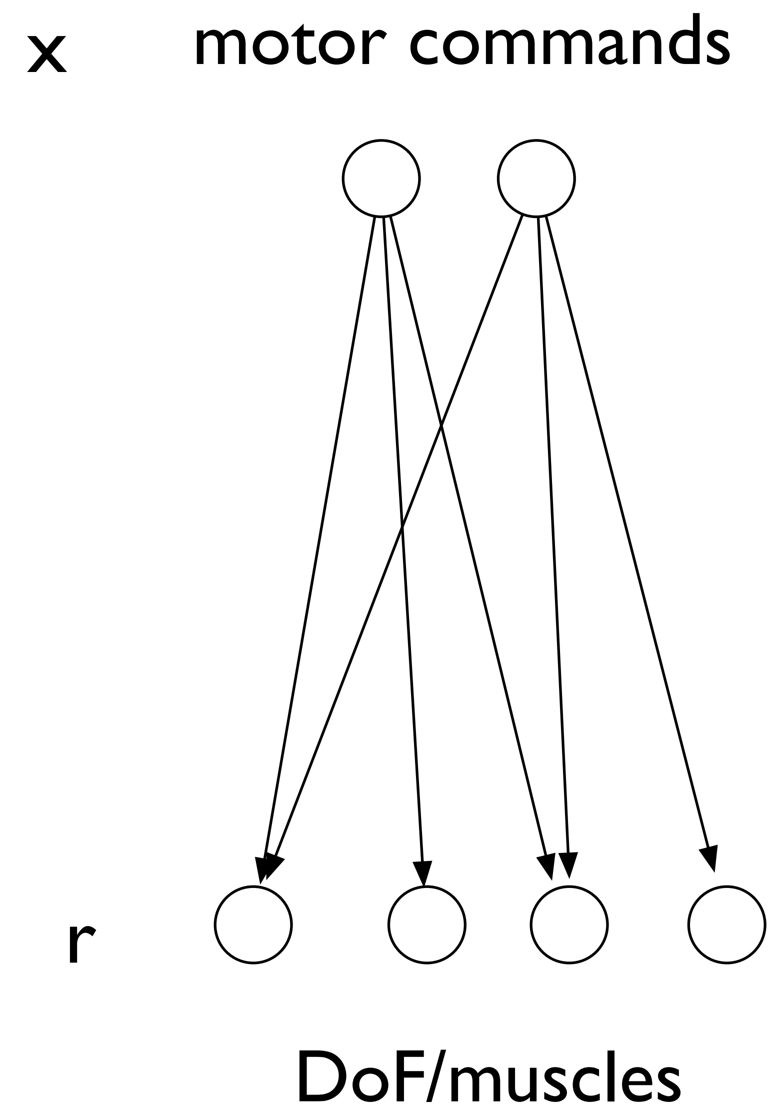


Bernstein's workers

- was the hammer position in space less variable than the joint configuration?
  - that is, does the task structure variance?
  - so that the solution to the degree of freedom problem lies in the variance/stability of the joint configuration?
- but: does this make any sense?
  - different reference frames for body vs. task
  - different units in the task vs joint space

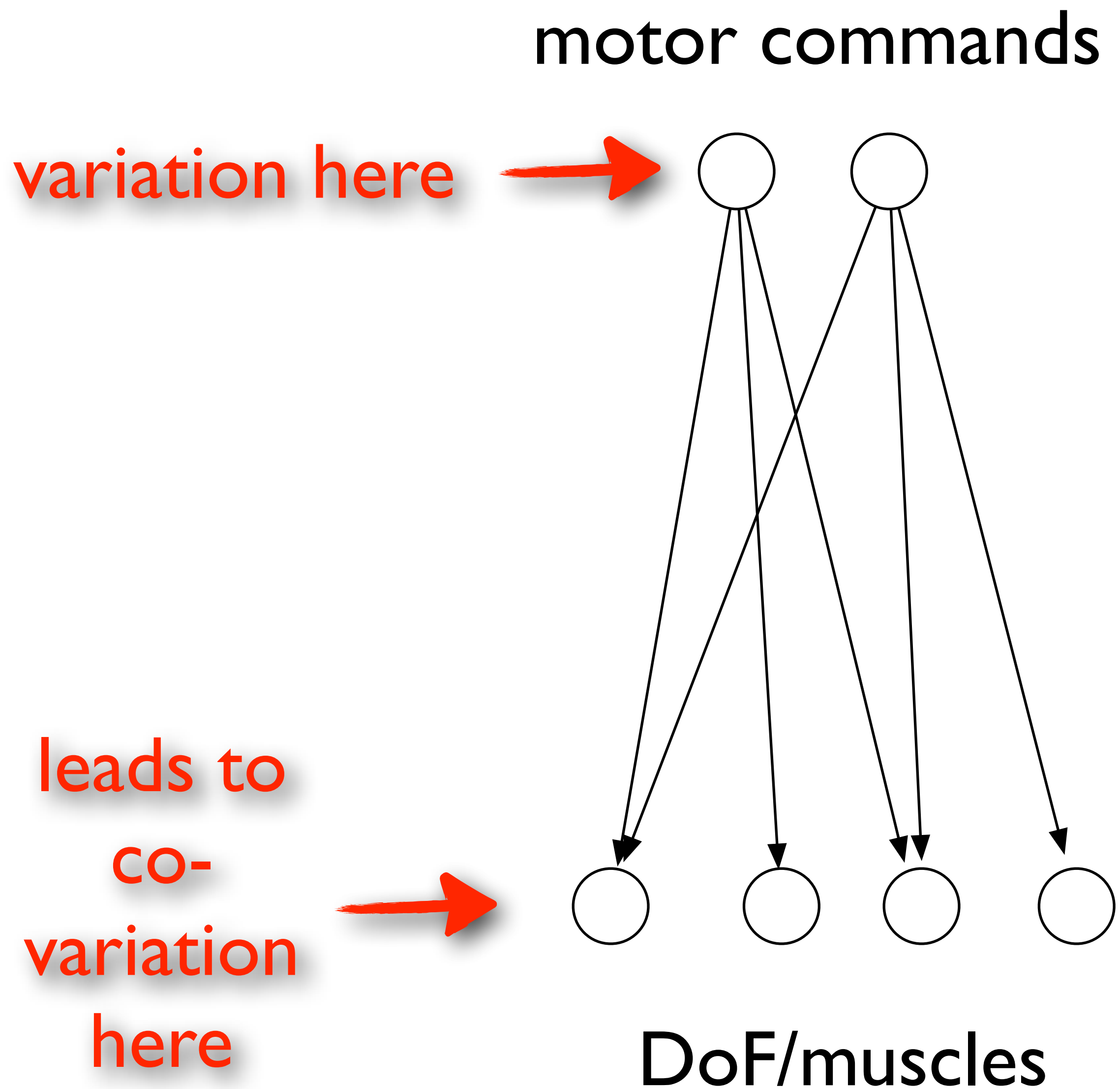
Classical synergy concept

- the task-level motor commands ‘x” activate synergies=groups of DoF through a forward neural network



Classical synergy concept

- command varies in time or across tasks => covariation of these muscle activations / DoF movements

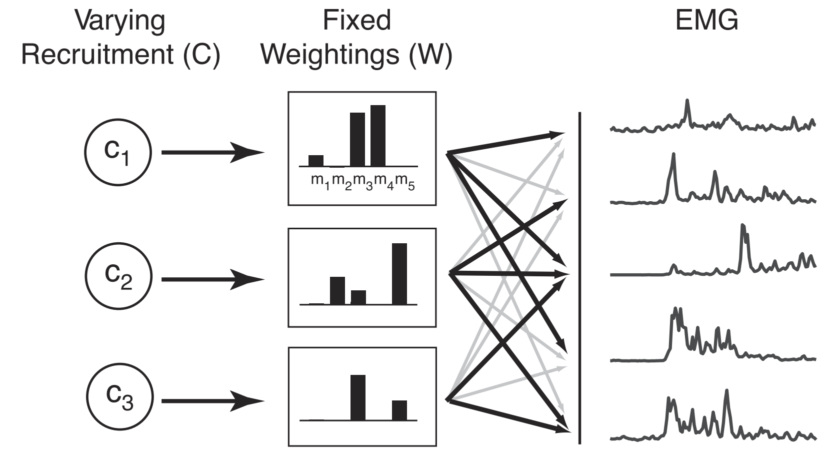


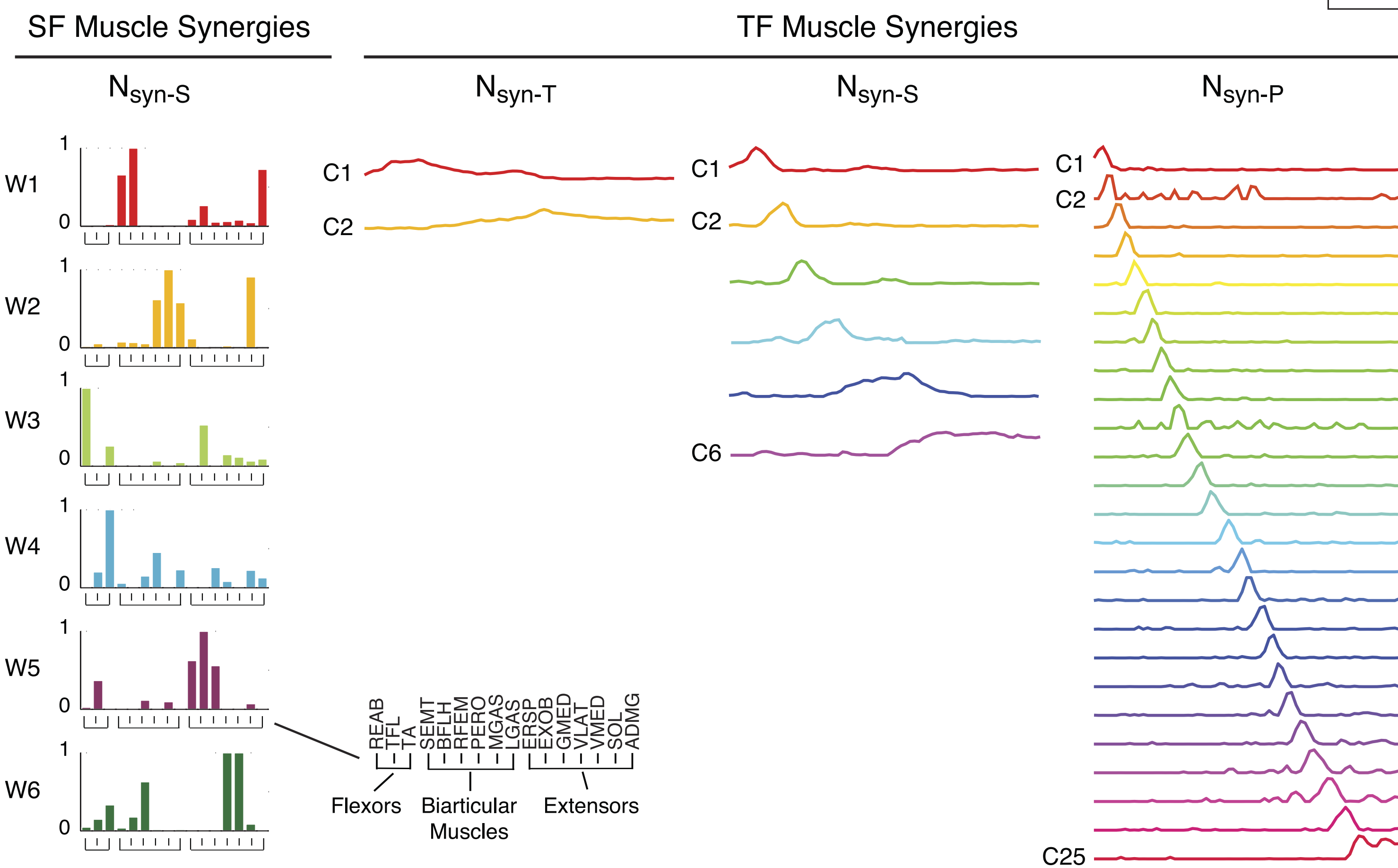
Classical synergy research strategy

- identify distinct synergies with the hope of finding a limited set => "the" synergies that explain multi-degree of freedom movement
- combine the time series of muscles/DoF under different conditions (sometimes including repetitions of movements) into one big data set and look for structure (e.g. principal components)
- if a small number of PC's is sufficient to account for most of the variance, conclude that few synergies at at work

Synergy: experimental use

- E.g, Safavynia, Ting, 2012:





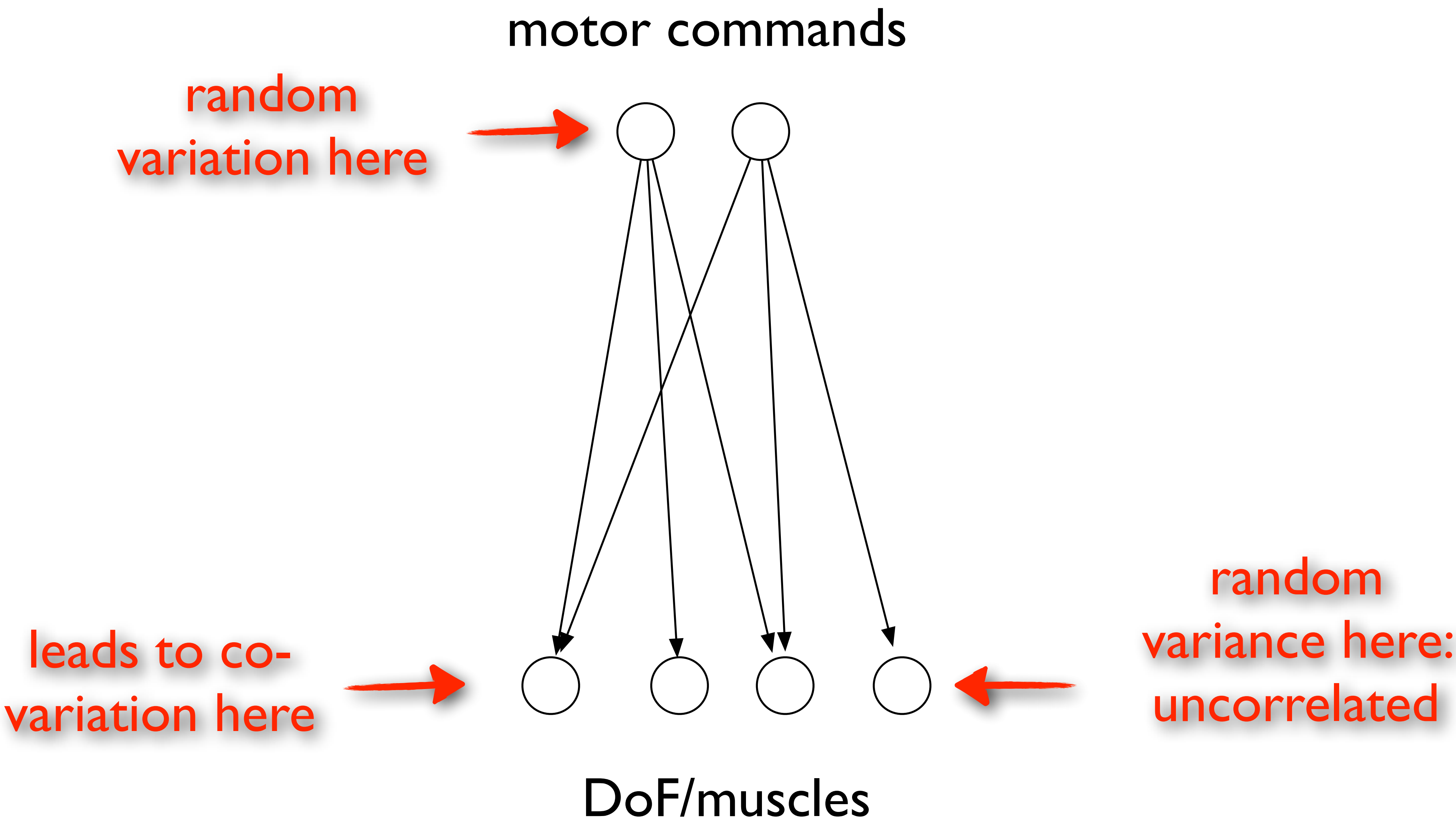
Classical synergy: critique of method

- … no invariant set of synergies has emerged
- confounds time, movement conditions, and trials
  - PCs are informative primarily about the geometry of the end-effector path.
  - and its variation with task
- [Steele, Tresch, Perreault: J Neurophysiol 2015]

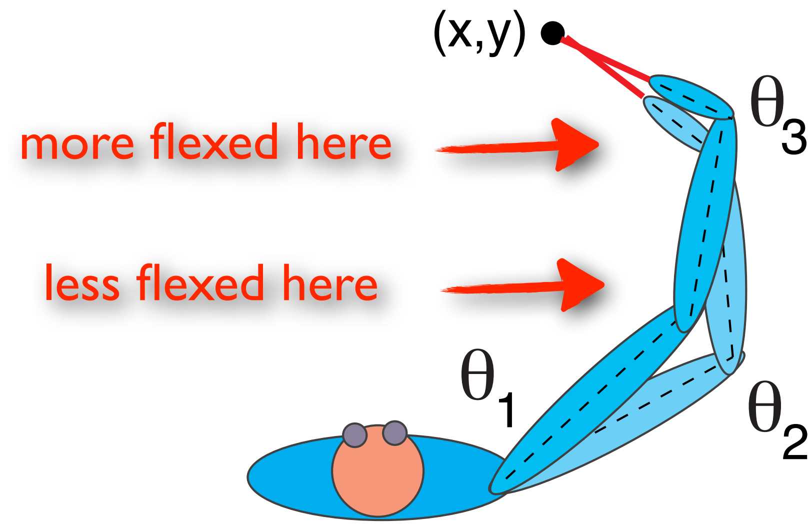
Classical synergy: critique of concept

- The variance across repetitions for a given task at given point in time = signature of stability
- That variance is structured in the OPPOSITE way than predicted!

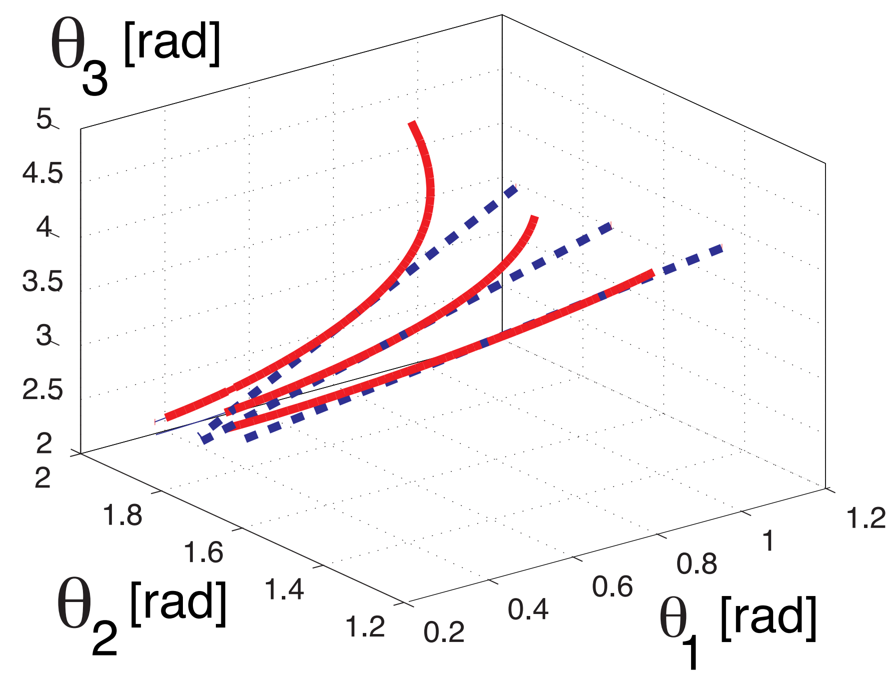
Classical synergy: critique of concept



Concept of the UnControlled Manifold



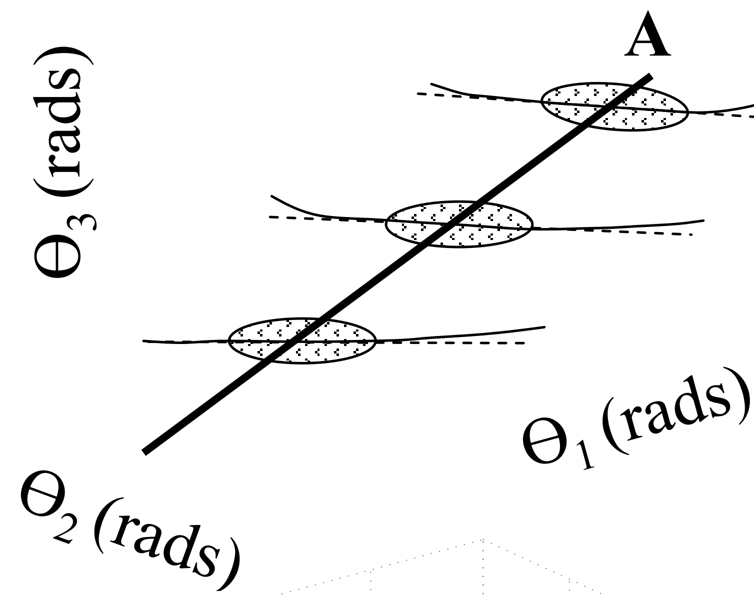
- the many DoF are coordinated such that changes that affect the task-relevant dimensions are resisted against more than changes that do not affect task relevant dimension
- leading to compensation

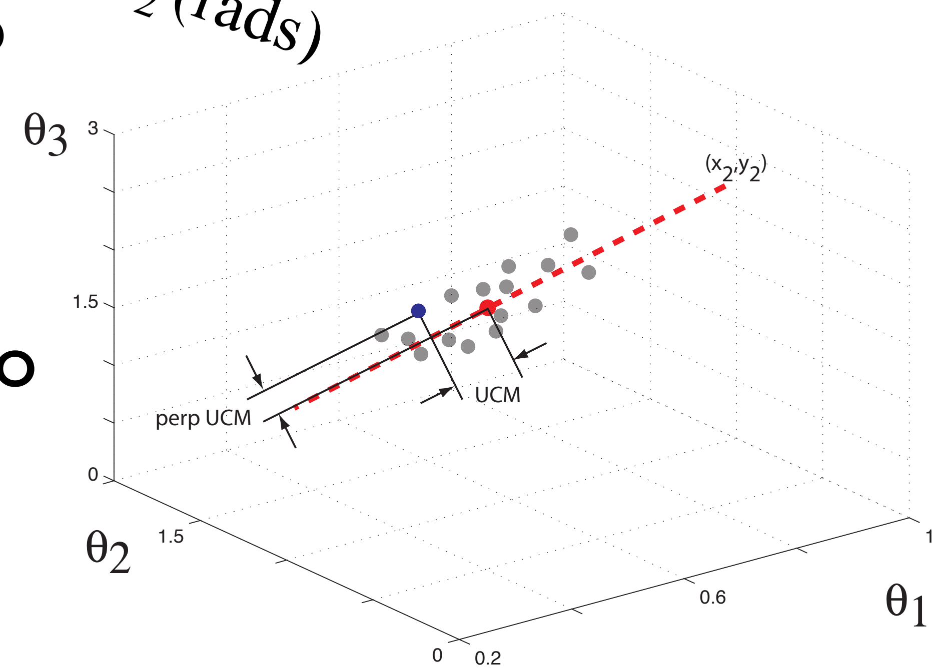


[Scholz, Schöner, EBR 126:289 (99)]

UCM synergy: data analysis

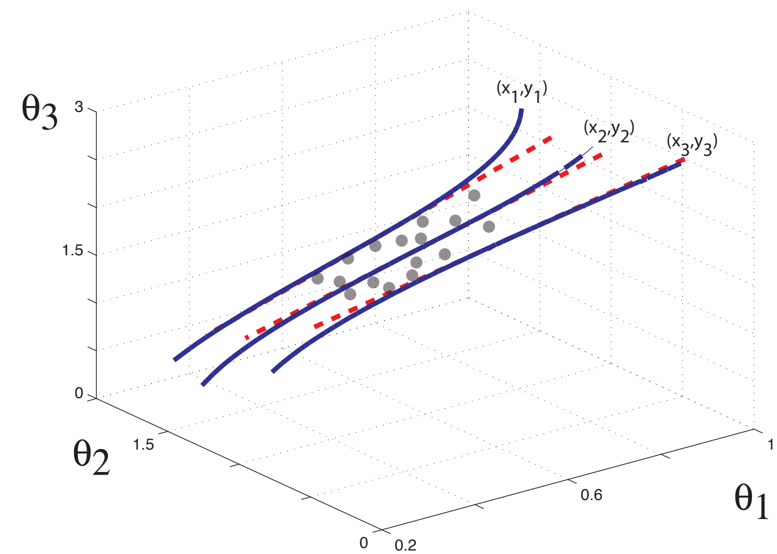
- align trials in time
- hypothesis about task variable
- compute null-space (tangent to the UCM)
- predict more variance within null space than perpendicular to it

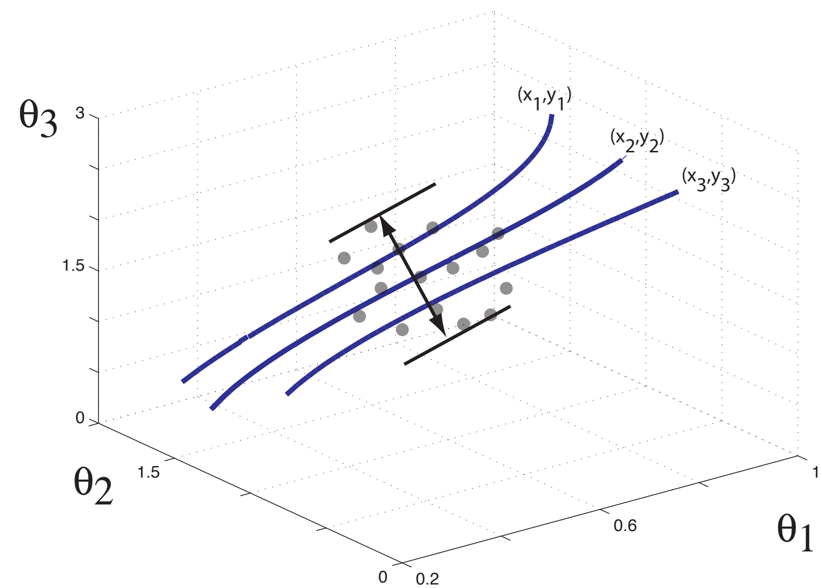




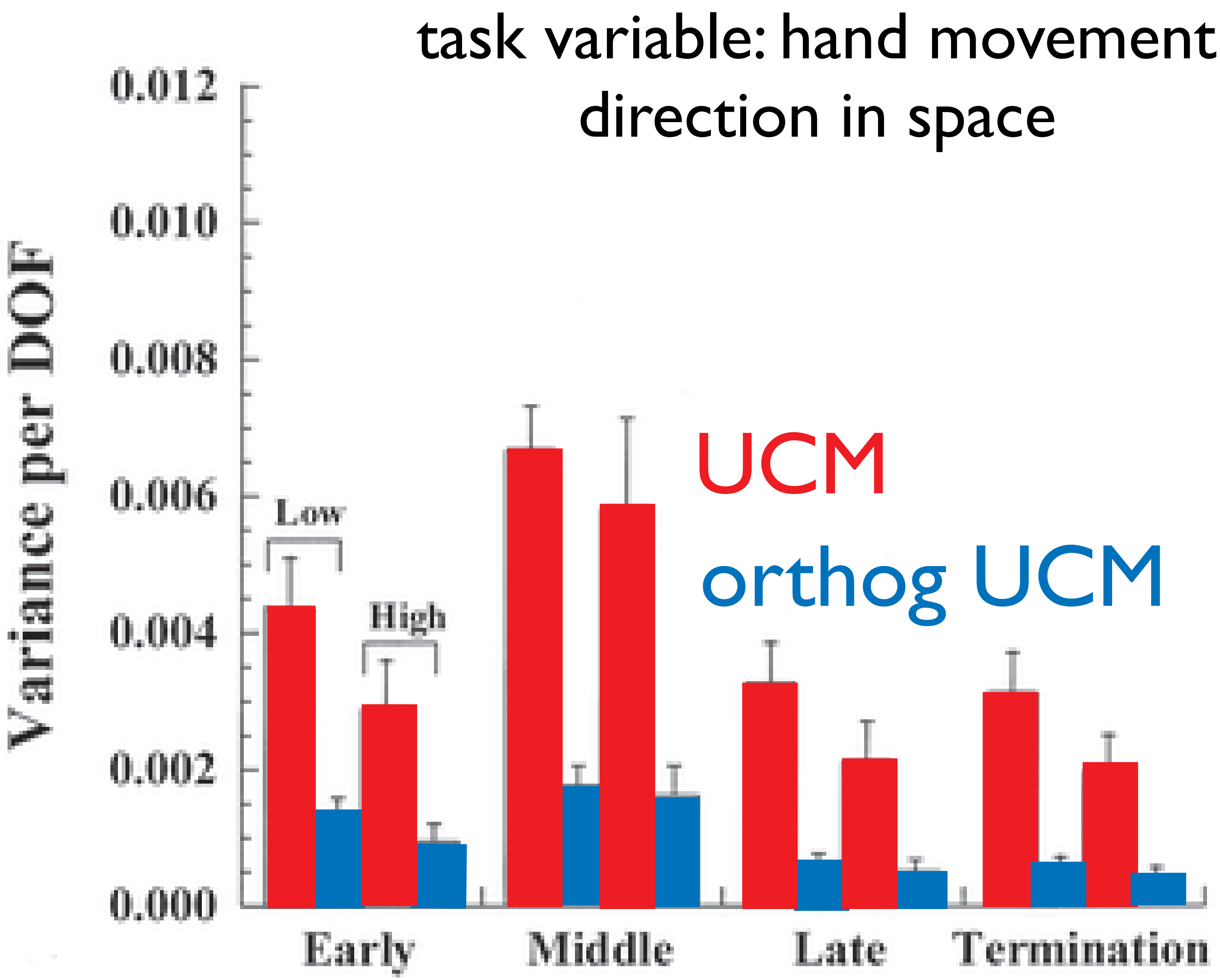
UCM synergy: data analysis

- supplement hypothesis testing by checking for correlation (Hermann, Sternad...)
  - look for increase in variance of task variable when correlation within data is destroyed

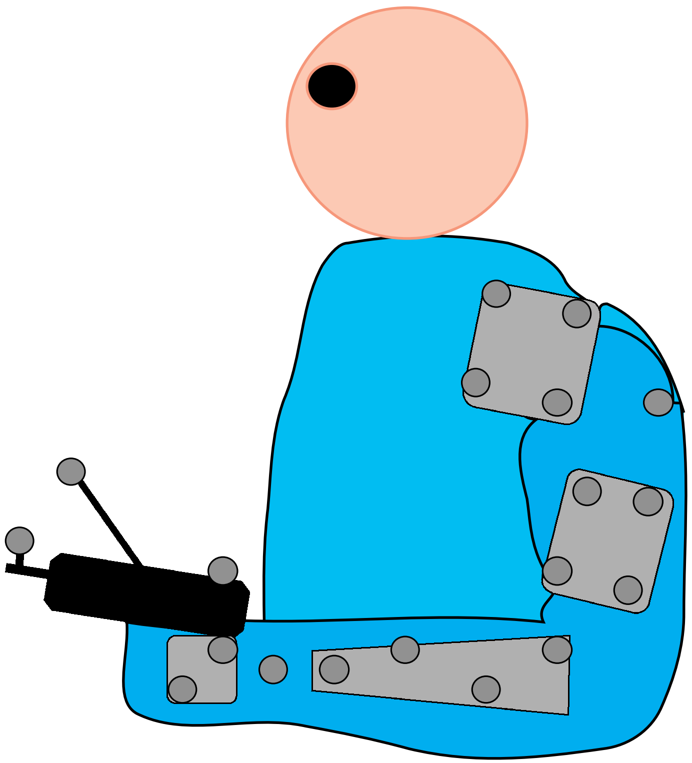


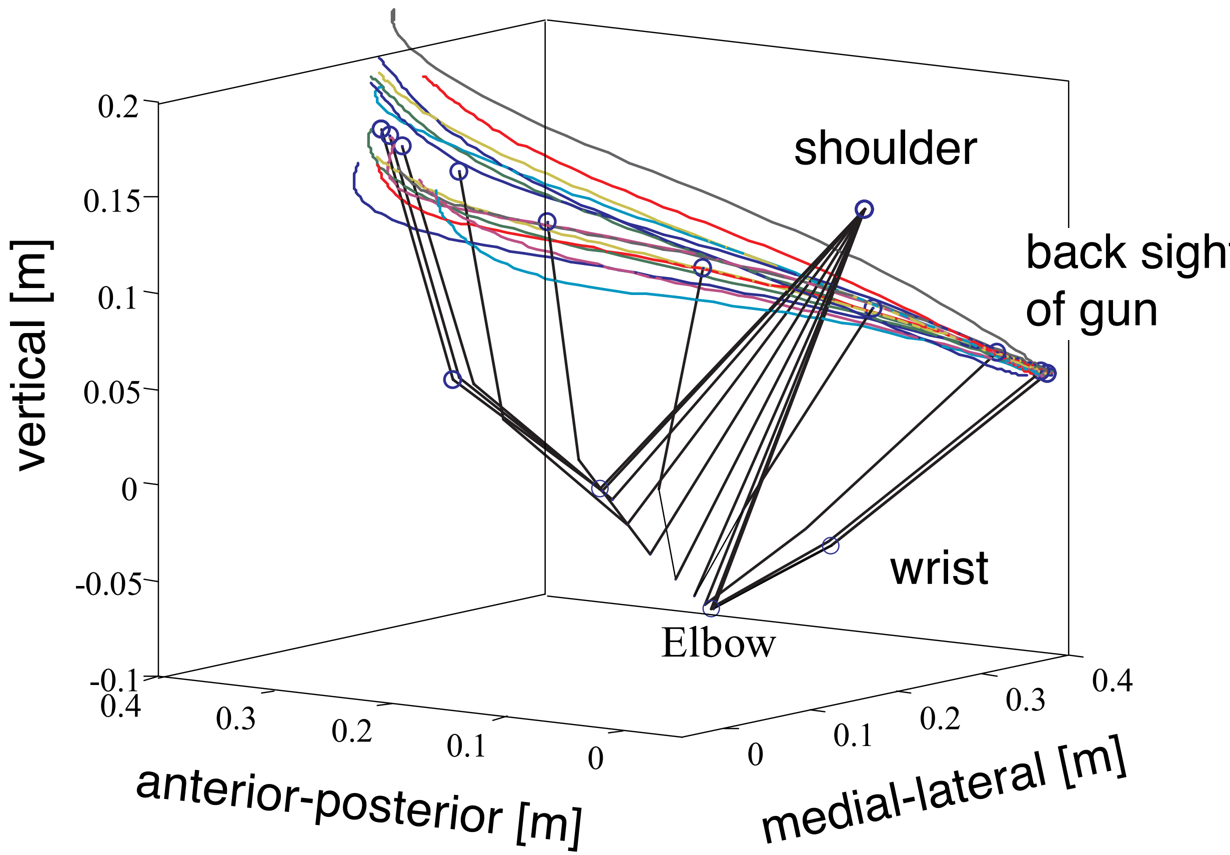


Example 1: pointing with 10 DoF arm at targets in 3D



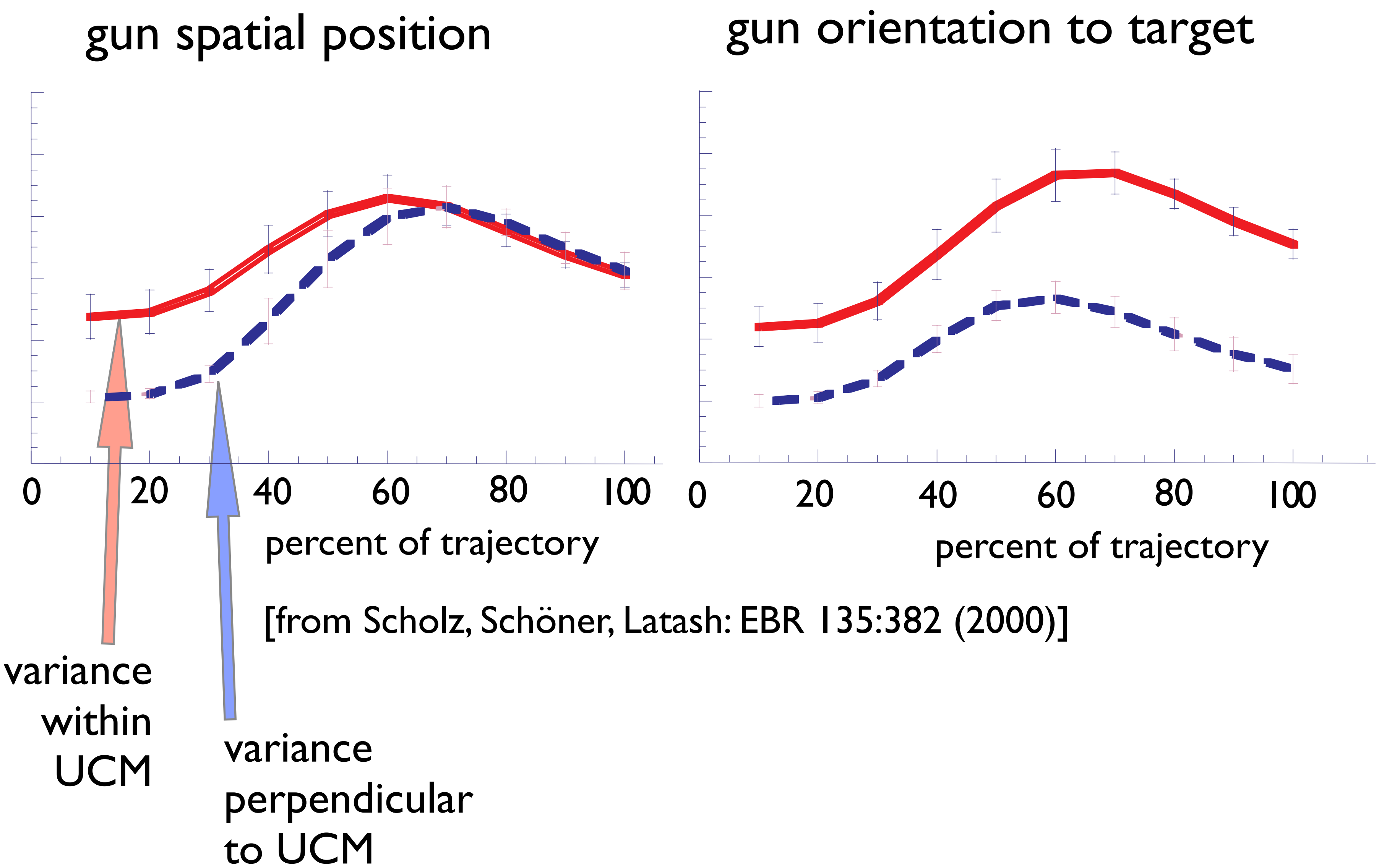
Example 2: shooting with 7 DoF arm at targets in 3D





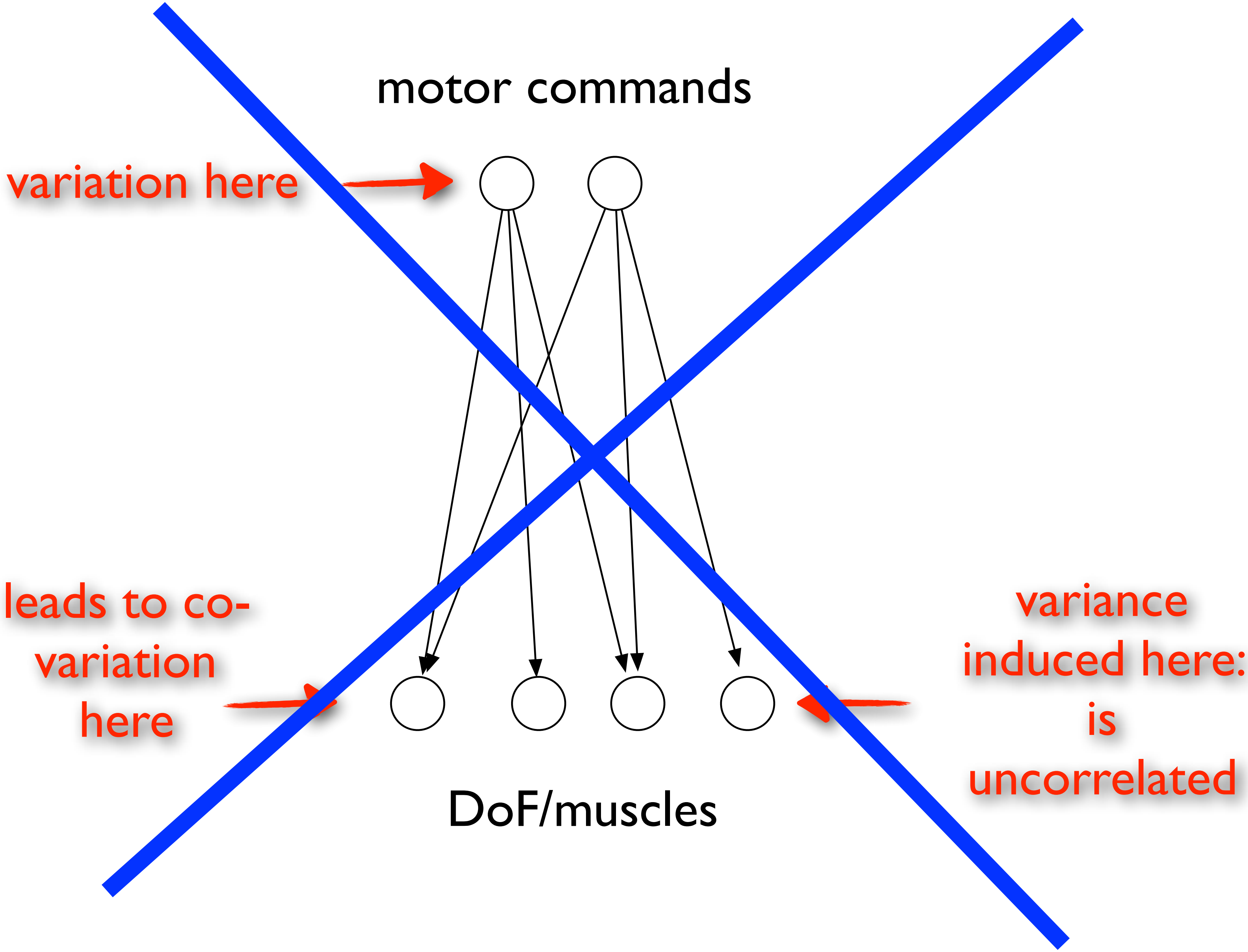
[from Scholz, Schöner, Latash: EBR 135:382 (2000)]

Example 2: shooting with 7 DoF arm at targets in 3D



[from Scholz, Schöner, Latash: EBR 135:382 (2000)]

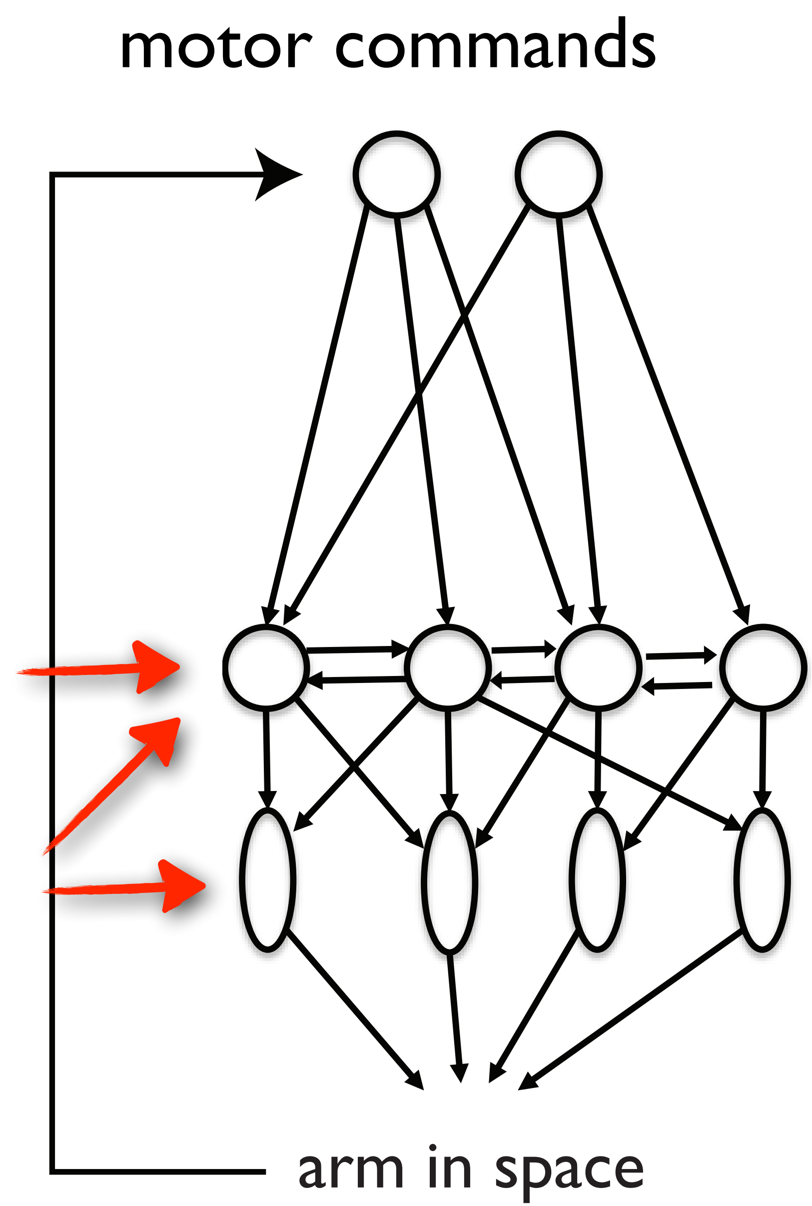
Synergy: critique of concept



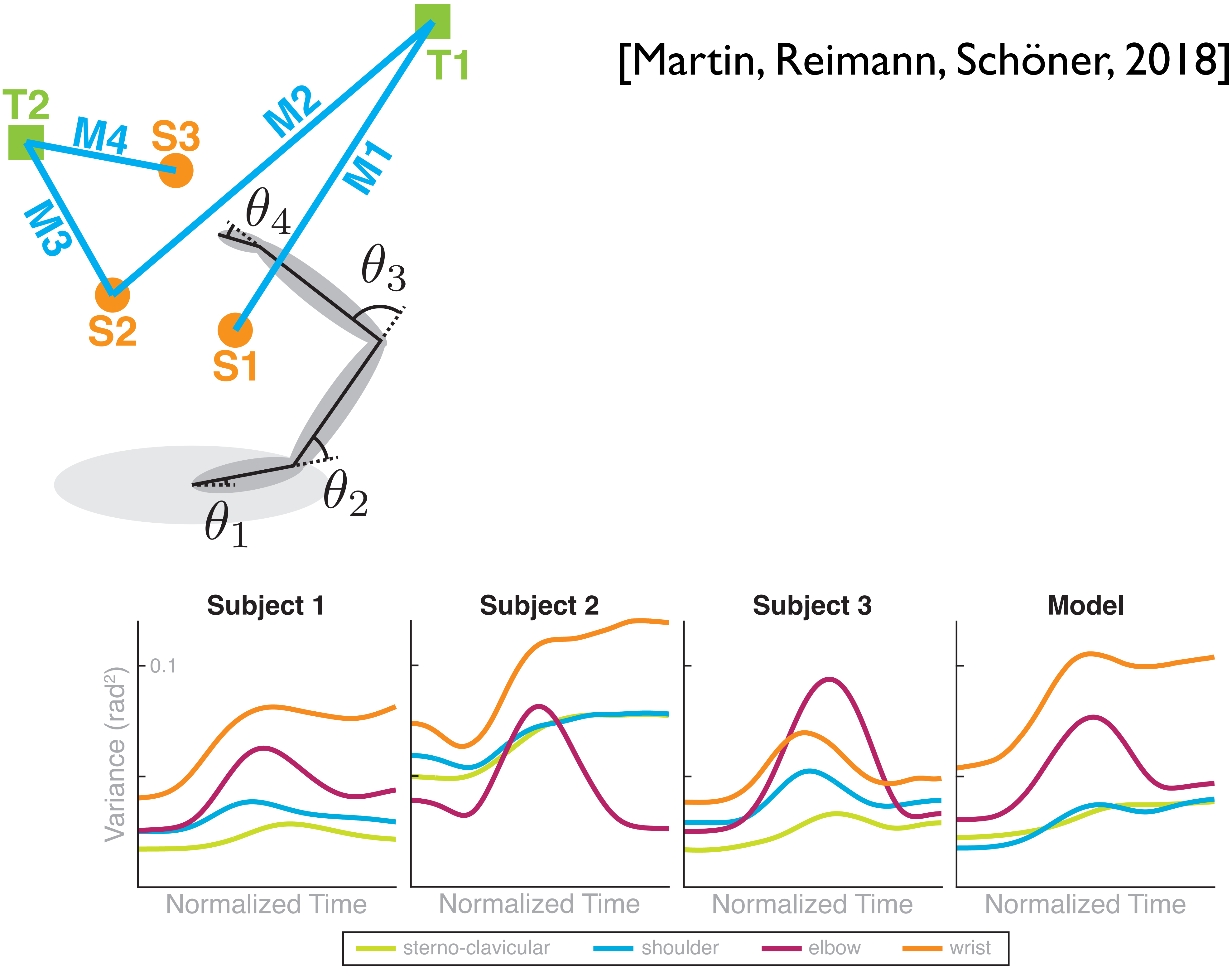
UCM synergy: decoupling

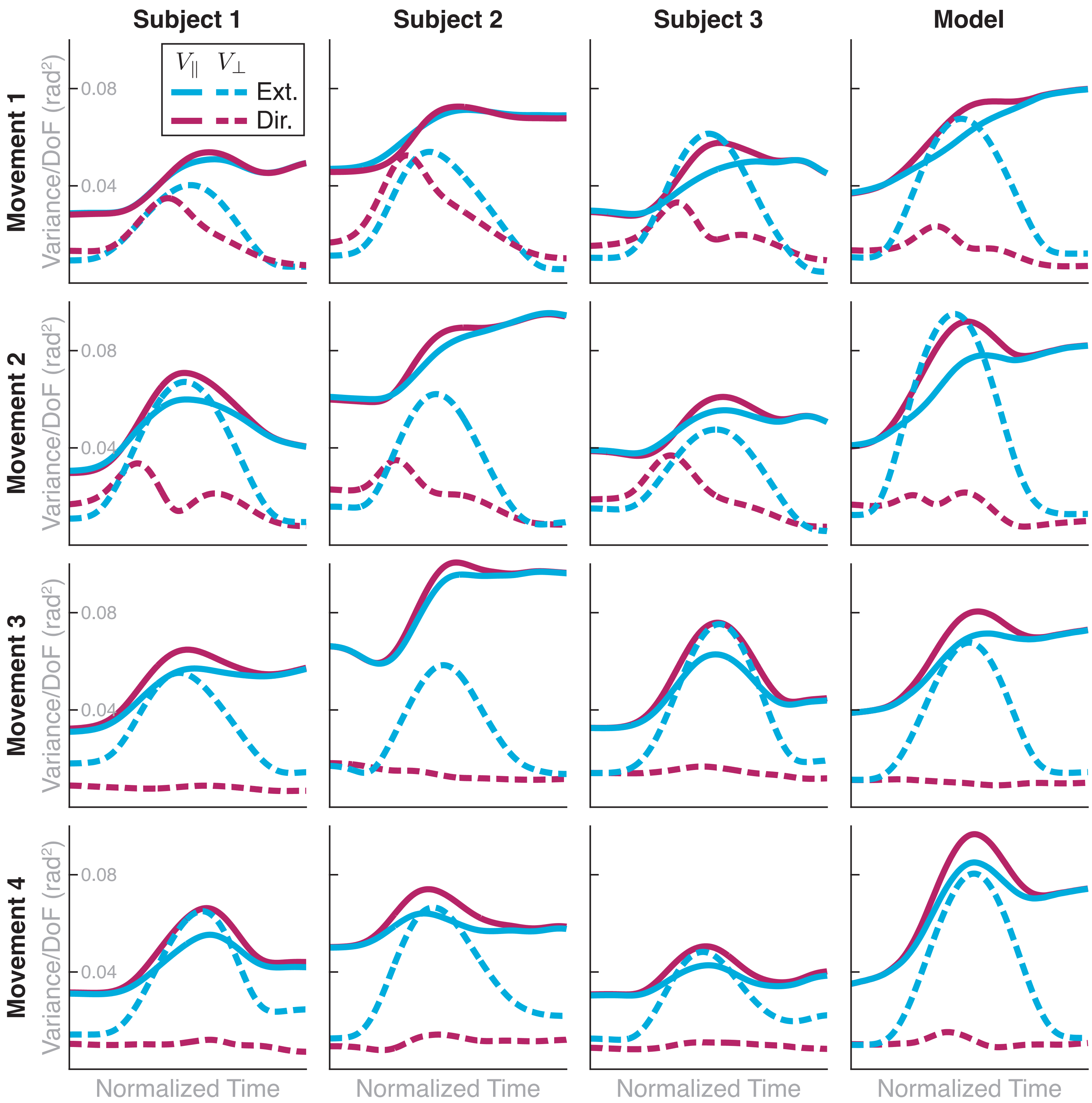
insert a perturbation here

compensatory change here



[Martin, Reimann, Schöner, 2018]





Subject 1
Subject 2
Subject 3
Model
$V_{\|}$ $V_{\perp}$
Ext.
Dir.
Movement 1
Movement 2
Movement 3
Movement 4
Variance/DoF (rad^2)
0.08
0.04
Normalized Time

model

biomechanical dynamics

$$M(\boldsymbol{\theta}) \cdot \ddot{\boldsymbol{\theta}} + H(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{T}_{\mathrm{m}}$$

muscle models

$$\begin{aligned} T_i &= K_l \cdot \left((e^{[K_{nl} \cdot (\theta_i - \lambda_i^p)]^+} - 1) - (e^{-[K_{nl} \cdot (\theta_i - \lambda_i^m)]^-} - 1) \right) \\ &+ \mu_{bl} \cdot \operatorname{asinh}(\dot{\theta}_i - \dot{\lambda}_i) + \mu_{rl} \cdot \dot{\theta}_i . \end{aligned}$$

neural dynamics of lambda

$$\dot{\mathbf{v}} = -\beta_v(\mathbf{v} - \mathbf{u}(t)),$$ timing signal

$$\mathbf{v}(t) = \mathbf{J}[\boldsymbol{\lambda}(t)] \cdot \dot{\boldsymbol{\lambda}}(t),$$

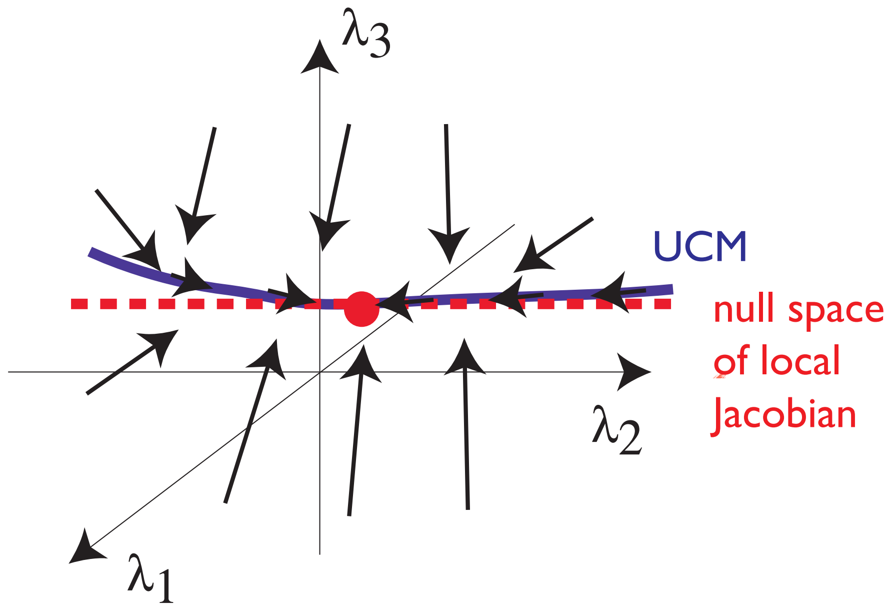
$$\ddot{\boldsymbol{\lambda}} = \left(\mathbf{J}^{+} \ \mathbf{E}\right) \cdot \begin{pmatrix} -\beta_v \mathbf{J} \cdot \dot{\boldsymbol{\lambda}} + \beta_v \mathbf{u} - \cancel{\dot{\mathbf{J}} \cdot \dot{\boldsymbol{\lambda}}} \\ -\beta_{s1}\mathbf{E}^T \cdot (\boldsymbol{\lambda} - \boldsymbol{\theta}_d) - \beta_{s2}\mathbf{E}^T \cdot (\dot{\boldsymbol{\lambda}} - \dot{\boldsymbol{\theta}}_d). - \cancel{\dot{\mathbf{E}}^T \cdot \dot{\boldsymbol{\lambda}}} \end{pmatrix}$$

back-
coupling

approximation

timing signal

$$\ddot{\boldsymbol{\lambda}} = \left(\mathbf{J}^+ \ \mathbf{E}\right) \cdot \begin{pmatrix} -\beta_v \mathbf{J} \cdot \dot{\boldsymbol{\lambda}} + \beta_v \mathbf{u} \\ 0 \end{pmatrix}$$



- => control is stable in range space
- => marginally stable in UCM/null space

where does this come from?

start with pseudo-inverse of: $v = J\dot{\lambda}$

$$\dot{\lambda} = J^{+}v$$

$$\ddot{\lambda} = J^{+}\dot{v} \quad [+\dot{J}^{+}v \approx 0]$$

a neuron, n, encoding rate of change of λ : $n = \dot{\lambda}$

$\dot{n} = J^{+}\dot{v}$ <= insert timing signal $\dot{v} = -v + u$

$\dot{n} = J^{+}(-v + u)$ <= insert $v = J\dot{\lambda}$

$\dot{n} = J^{+}(-J\dot{\lambda} + u)$ <= replace $n = \dot{\lambda}$

$$\dot{n} = J^{+}(-Jn + u)$$

$$\dot{n} = -J^{+}Jn + J^{+}u$$

where does this come from?

$$\dot{n} = -J^+ J n + J^+ u$$

$$\dot{n} = -n + n - J^+ J n + J^+ u$$

$$\dot{n} = -n + (1 - J^+ J) n + J^+ u$$

projection onto null-space

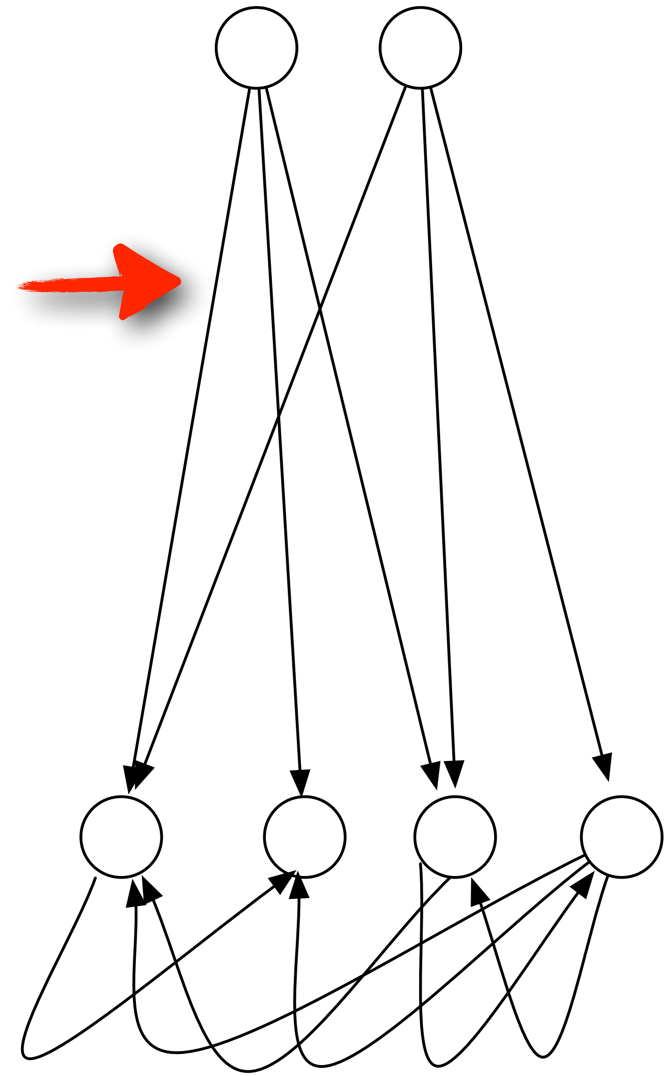
feed-forward from timing command

where does this come from?

feed-
forward
from timing
command

$$\dot{n} = -n + (1 - J^{+}J)n + J^{+}u$$

projection
onto null-
space



how does this do the UCM effect?

projection onto null-space

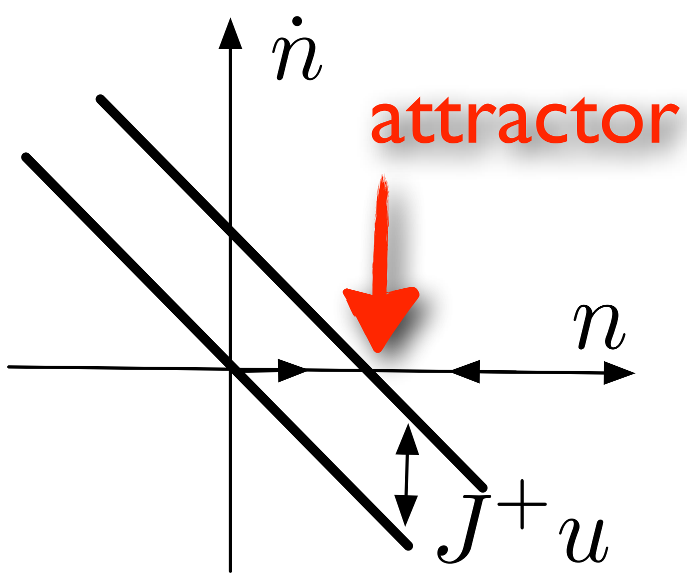
feed-forward from timing command

$$\dot{n} = -n + (1 - J^{+}J)n + J^{+}u$$

within the range-space

$$\dot{n} = -n + J^{+}u$$

=> stability within the range-space



how does this do the UCM effect?

projection onto null-space

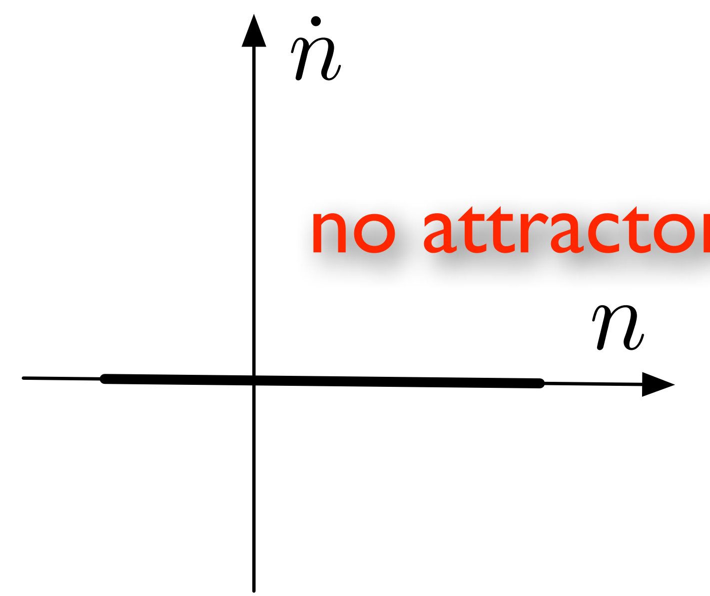
feed-forward from timing command

$$\dot{n} = -n + (1 - J^+ J)n + J^+ u$$

within the null-space

$$\dot{n} = -n + n + 0$$

$$\dot{n} = 0$$



=> no stability within the null-space

.... to be continued

- sorry for the abrupt shift in level of difficulty…