Gaussian Process II:
Learning hyper parameters

Choosing kernel parameters

$$k(t_i, t_j) = \sigma_f^2 \exp\left\{-\frac{1}{2\ell^2}(t_i - t_j)^2\right\}$$

- Choose some *hyperparameters*: $\sigma_f = 14$, $\ell = 50$

$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i, t_j)\}_{i,j} = \begin{bmatrix} 196 & 26.5 & 00.0 \\ 26.5 & 196 & 0.01 \\ 00.0 & 0.01 & 196 \end{bmatrix}$$

- Choose some *hyperparameters*: $\sigma_f = 7$, $\ell = 100$

$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i, t_j)\}_{i,j} = \begin{bmatrix} 49.0 & 29.7 & 00.2 \\ 29.7 & 49.0 & 03.6 \\ 00.2 & 03.6 & 49.0 \end{bmatrix}$$

- Choose some *hyperparameters*: $\sigma_f = 7$, $\ell = 500$

$$t = \begin{bmatrix} 0700 \\ 0800 \\ 1029 \end{bmatrix} \qquad K(t,t) = \{k(t_i, t_j)\}_{i,j} = \begin{bmatrix} 49.0 & 48.0 & 39.5 \\ 48.0 & 49.0 & 44.1 \\ 39.5 & 44.1 & 49.0 \end{bmatrix}$$

Sampling from a Covariance matrix

Scalar case

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

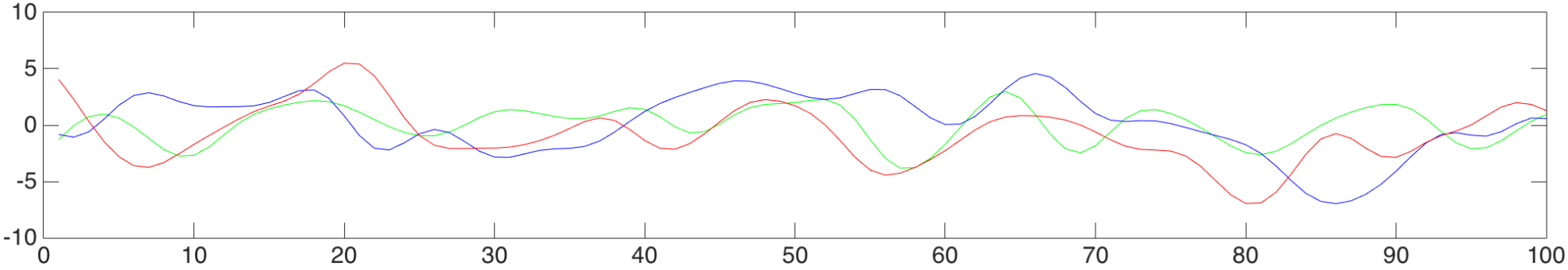
$$x \sim \mu + \sigma(\mathcal{N}(0, 1)$$

Vector case in GP setting

$$f_* \sim \mu + B(\mathcal{N}(0, I)$$

$$BB^T = \Sigma_*$$

Cholesky decomposition accomplishes this factorization where are B and B^T upper and lower triangular matrices




```
function [ Km ] = GaussianF( numPts,s,l )
%Priors for GP ; numPts=100, s =2, l=3

Km = zeros(numPts,numPts);
for i = 1: numPts
    for j = 1: numPts
        Km(i,j) = (s^2)*exp(-(0.5/l^2)*(i-j)^2);
    end
end
end
```

```
F=chol(Km)'*randn(100,1);
```

If f and y are jointly Gaussian:

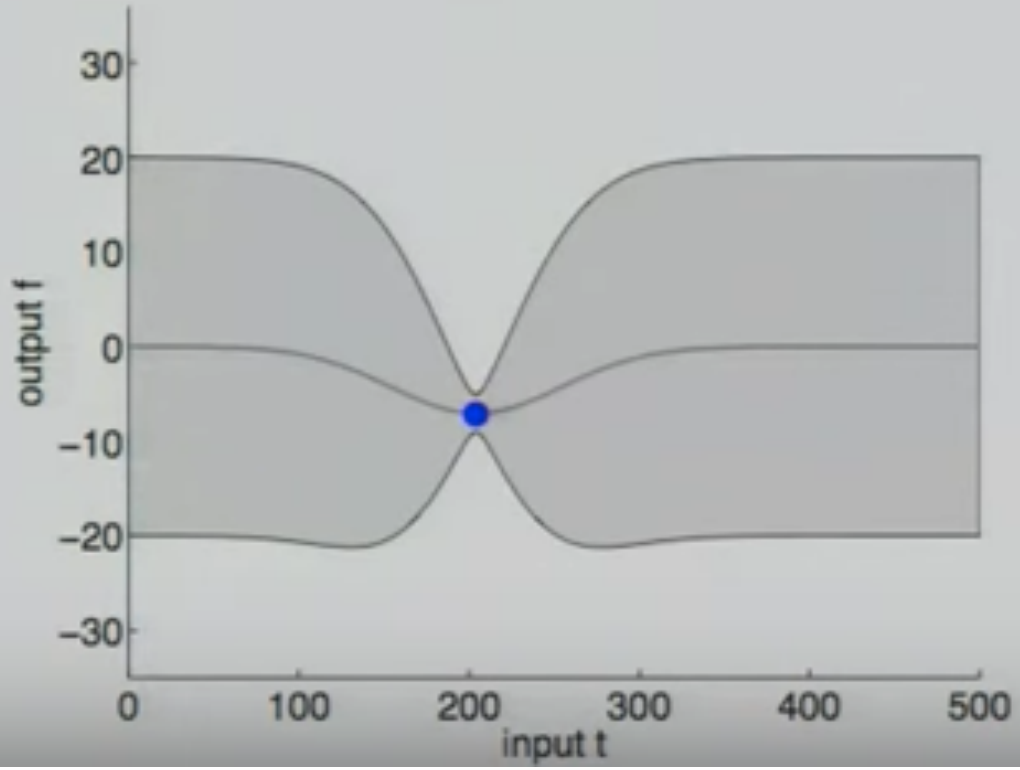
$$\begin{bmatrix} f \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_f \\ m_y \end{bmatrix}, \begin{bmatrix} K_{ff} & K_{fy} \\ K_{fy}^T & K_{yy} \end{bmatrix} \right)$$

Then:

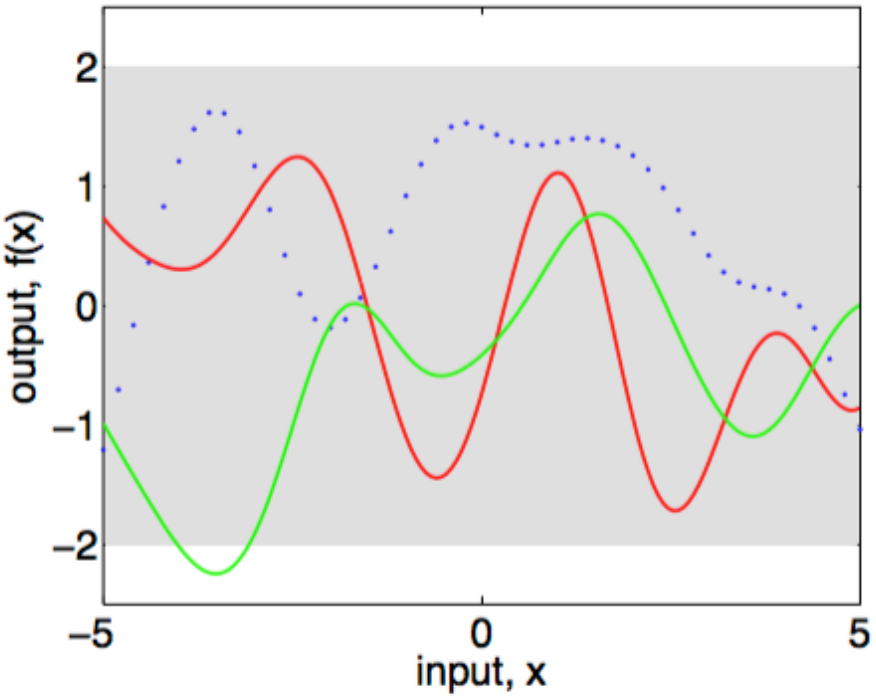
$$f|y \sim \mathcal{N}\left(K_{fy} K_{yy}^{-1} (y - m_y) + m_f \;\;,\;\; K_{ff} - K_{fy} K_{yy}^{-1} K_{fy}^T \right)$$

- Use conditioning to update the posterior:

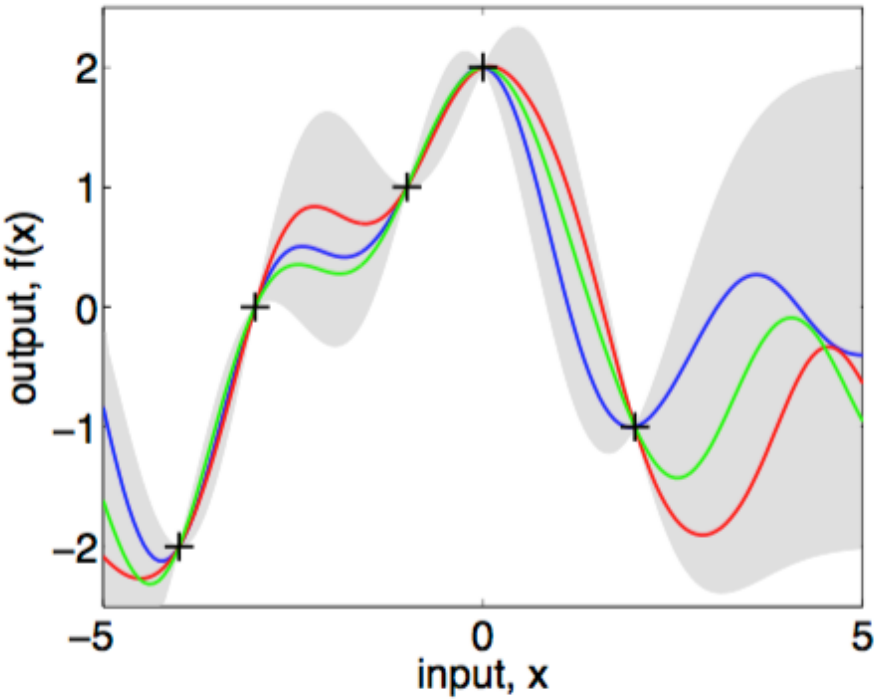
$$f|y(204) \sim \mathcal{N}\left(K_{fy}K_{yy}^{-1}(y(204) - m_y) \;,\; K_{ff} - K_{fy}K_{yy}^{-1}K_{fy}^{T}\right)$$



An impractical but intuitively appealing way to think about
Bayesian reasoning in the GP in the noiseless case:
Of all the possible functions, just keep the ones that fit the data

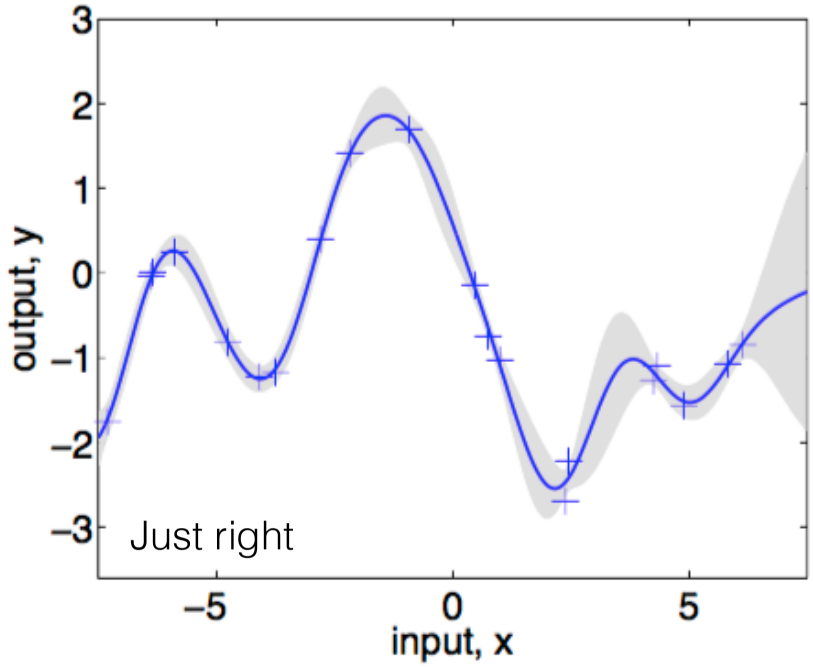


(a), prior

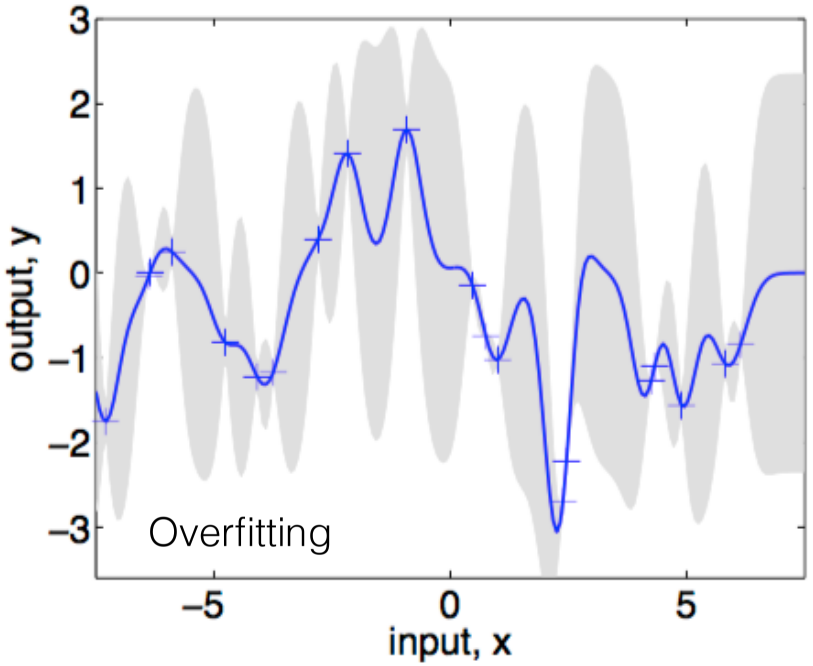


(b), posterior

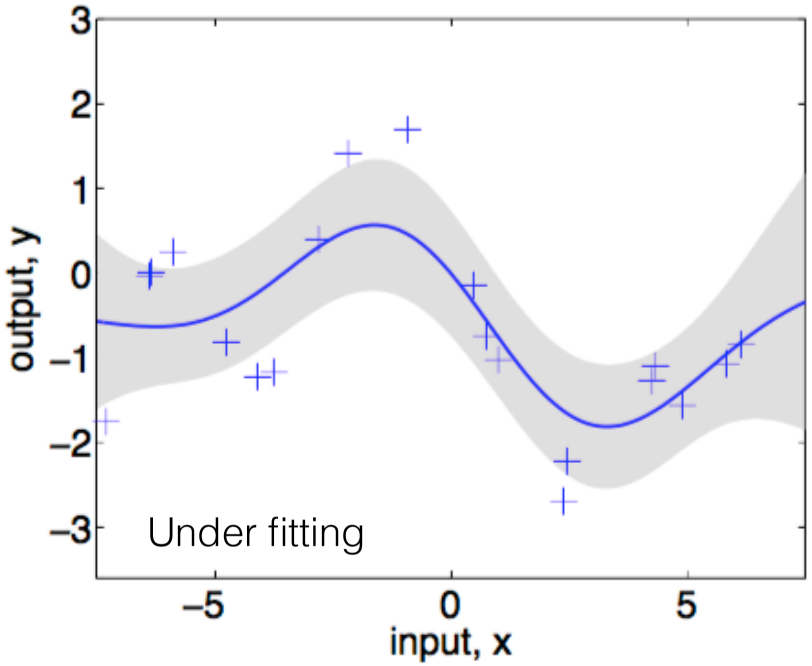
A practical problem: choosing the best set of hyperparameters



(a), $\ell = 1$



(b), $\ell = 0.3$



(c), $\ell = 3$

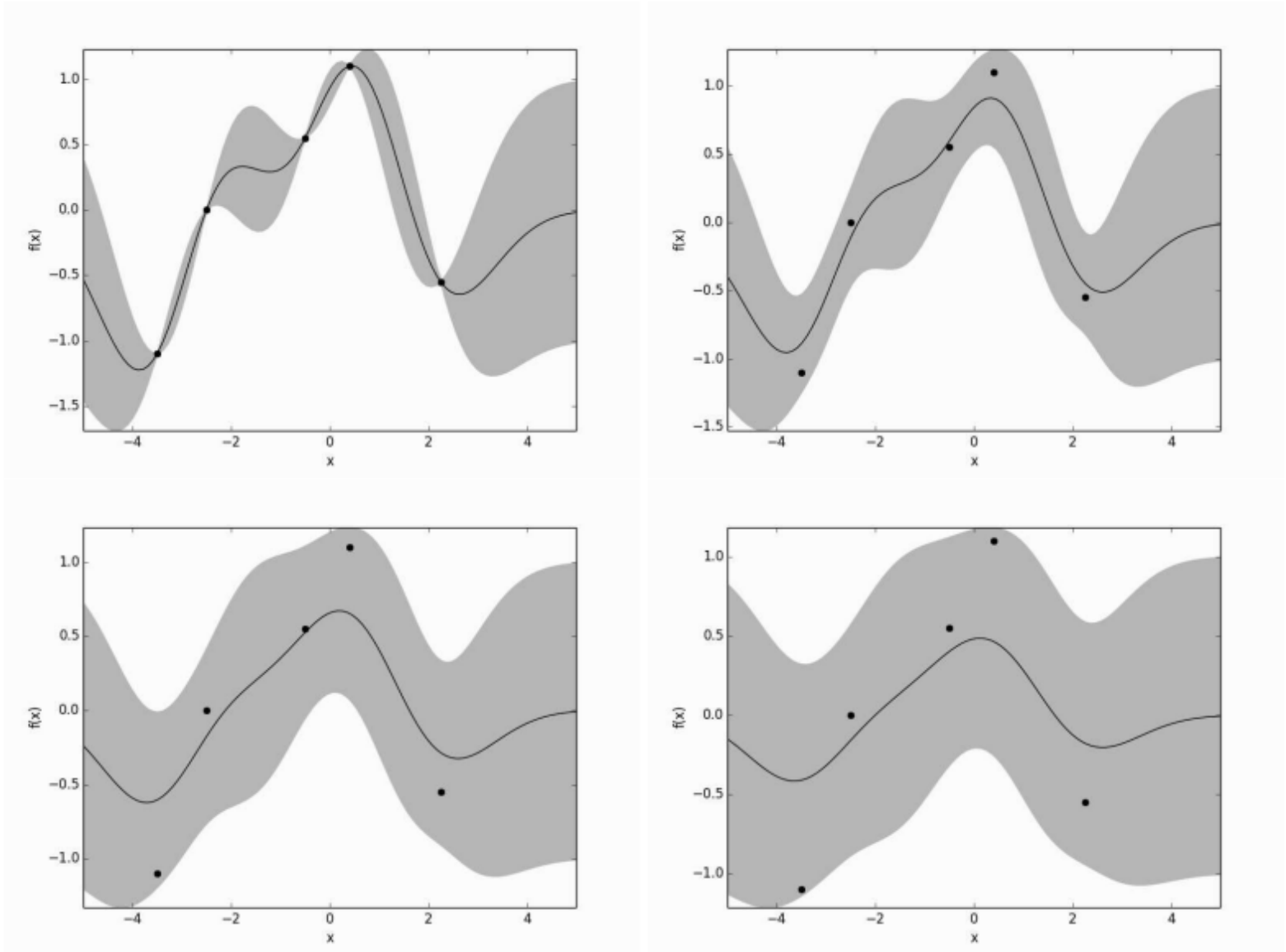


FIGURE 18.4 The effects of adding noise to the estimate of the covariance in the training data with the squared exponential kernel. Each plot shows the mean and 2 standard deviation error bars for a Gaussian process fitted to the five datapoints marked with dots. *Top left:* $\sigma_n = 0.0$, *top right:* $\sigma_n = 0.2$, *bottom left:* $\sigma_n = 0.4$, *bottom right:* $\sigma_n = 0.6$

Now for learning hyper parameters …

$$P(\mathbf{f}|\mathbf{X}) = (2\pi)^{\frac{n}{2}} |\mathbf{K}|^{\frac{1}{2}} \exp(-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f})$$

Simpler to work with log likelihood …

$$\mathbf{f}|\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$$

$$\log p(\mathbf{f}|X) = -\tfrac{1}{2}\mathbf{f}^\top K^{-1}\mathbf{f} - \tfrac{1}{2}\log|K| - \tfrac{n}{2}\log 2\pi$$

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K + \sigma_n^2 I)$$

$$\log p(\mathbf{y}|X) = -\tfrac{1}{2}\mathbf{y}^\top (K + \sigma_n^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K + \sigma_n^2 I| - \tfrac{n}{2}\log 2\pi$$

Marsland's notation: $\mathbf{t}$ instead of $\mathbf{y}$

$$\log P(\mathbf{t}|\mathbf{x},\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{t}^T(\mathbf{K}+\sigma_n^2\mathbf{I})^{-1}\mathbf{t} - \frac{1}{2}\log|\mathbf{K}+\sigma_n^2\mathbf{I}| - \frac{N}{2}\log 2\pi$$

$$\mathbf{Q} = (\mathbf{K}+\sigma_n^2\mathbf{I})$$

Take derivatives wrt a parameter θ

$$\begin{aligned}\frac{\partial \mathbf{Q}^{-1}}{\partial \boldsymbol{\theta}} &= -\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}}\mathbf{Q}^{-1} \\ \frac{\partial \log|\mathbf{Q}|}{\partial \boldsymbol{\theta}} &= \text{trace}\left(\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}}\right)\end{aligned}$$

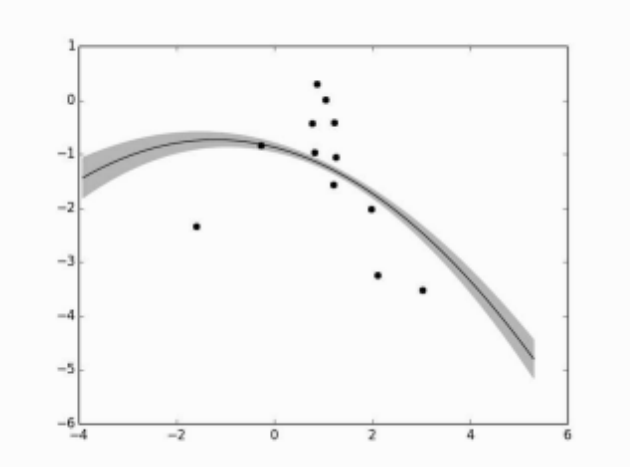
$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= \exp(\sigma_f) \exp\left(-\frac{1}{2}\exp(\sigma_l)|\mathbf{x} - \mathbf{x}'|^2\right) + \exp(\sigma_n)\mathbf{I} \\
&= k' + \exp(\sigma_n)\mathbf{I}
\end{aligned}
$$

(Use exp(parameter) to make sure they are positive

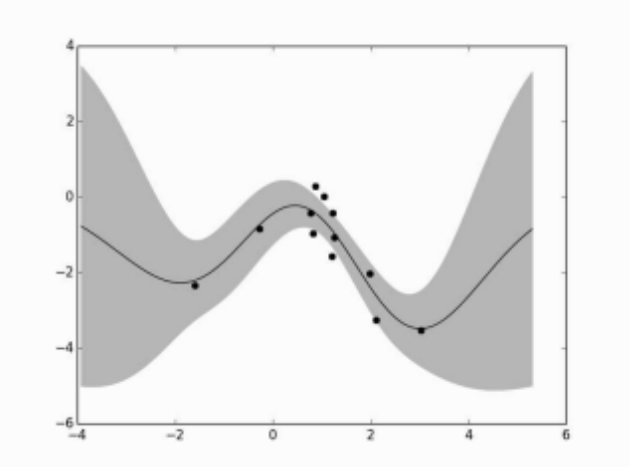
$$
\frac{\partial k}{\partial \sigma_f} = k'
$$

$$
\frac{\partial k}{\partial \sigma l} = k' \times \left(-\frac{1}{2}\exp(\sigma_l)|\mathbf{x} - \mathbf{x}'|^2\right)
$$

$$
\frac{\partial k}{\partial \sigma_n} = \exp(\sigma_n)\mathbf{I}
$$



Initial condition



fitted parameters

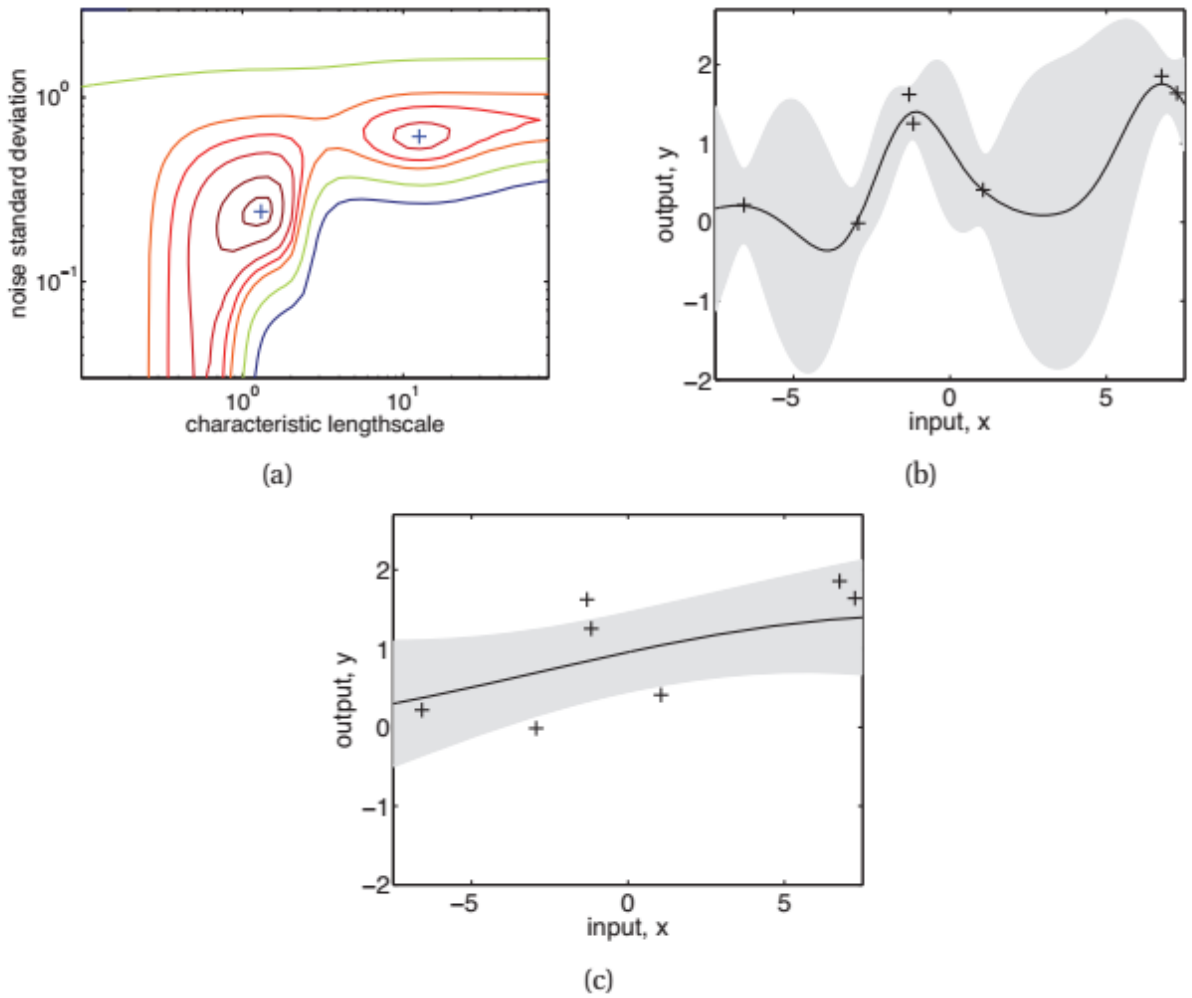
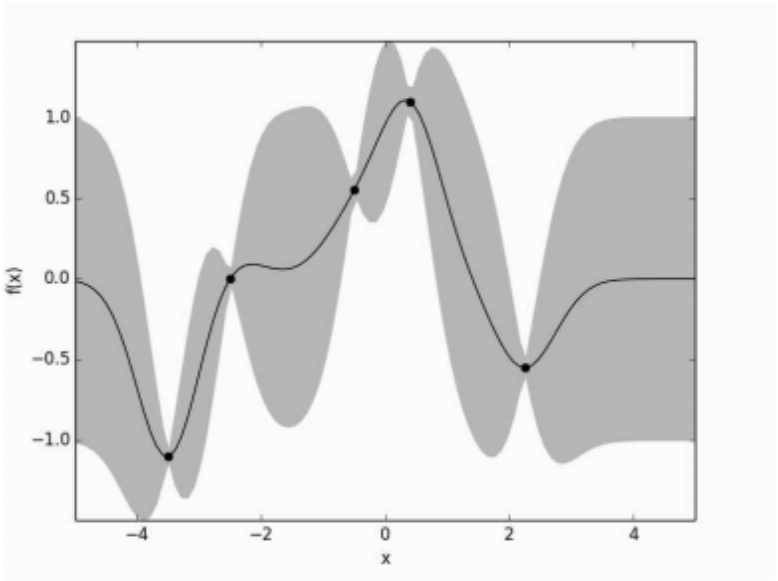
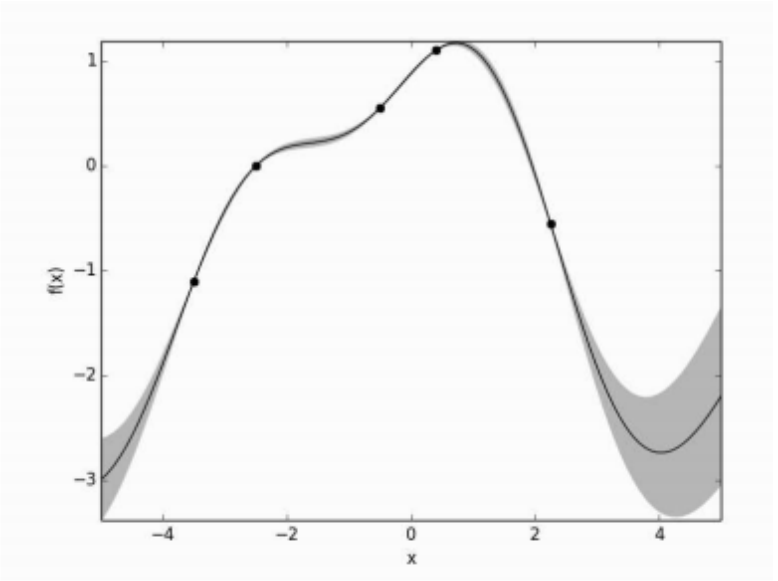


Figure 15.5 Illustration of local minima in the marginal likelihood surface. (a) We plot the log marginal likelihood vs σ_y^2 and ℓ, for fixed $\sigma_f^2 = 1$, using the 7 data points shown in panels b and c. (b) The function corresponding to the lower left local minimum, $(\ell, \sigma_n^2) \approx (1, 0.2)$. This is quite "wiggly" and has low noise. (c) The function corresponding to the top right local minimum, $(\ell, \sigma_n^2) \approx (10, 0.8)$. This is quite smooth and has high noise. The data was generated using $(\ell, \sigma_n^2) = (1, 0.1)$. Source: Figure 5.5 of (Rasmussen and Williams 2006). Figure generated by `gprDemoMarglik`, written by Carl Rasmussen.



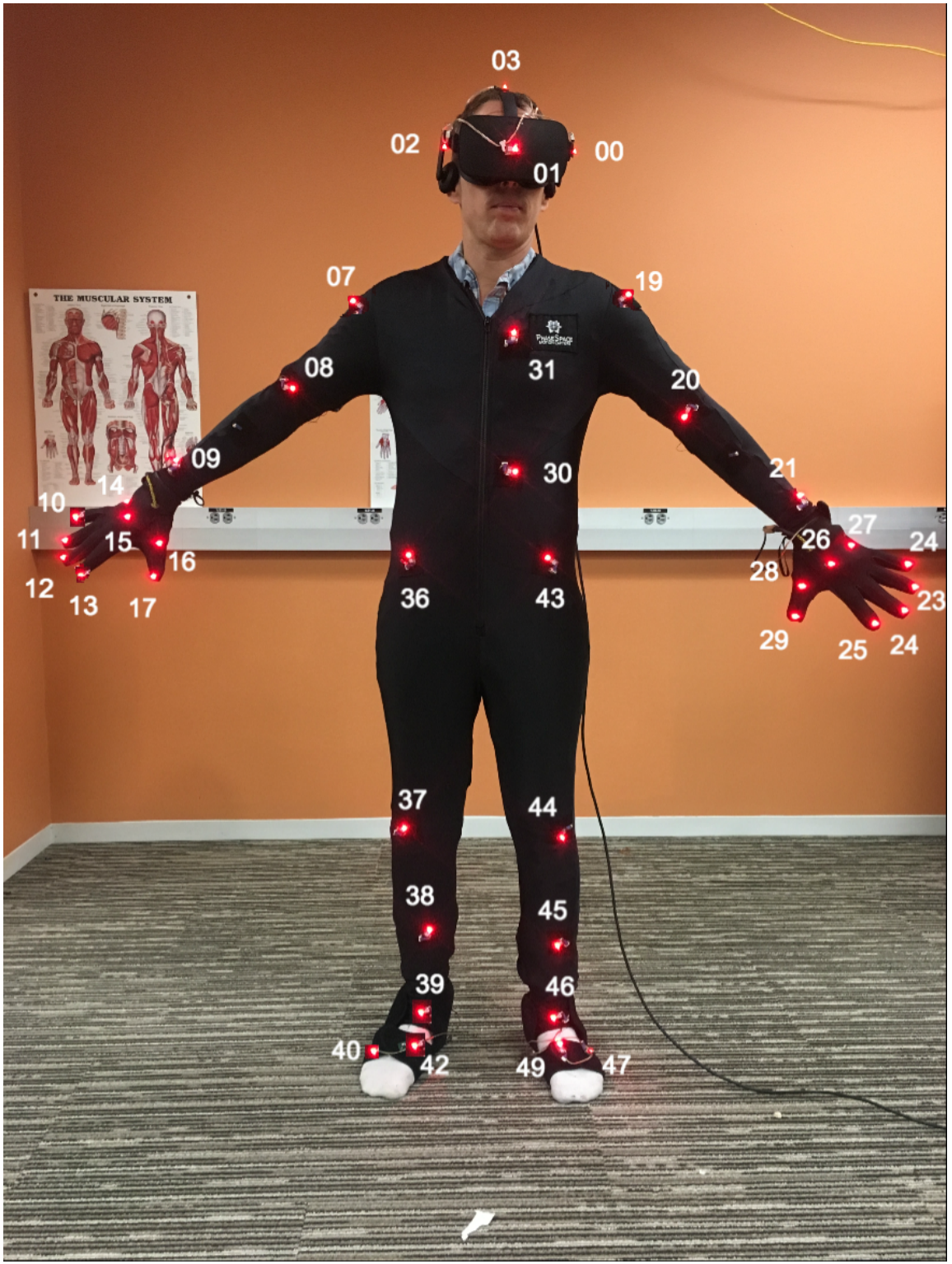


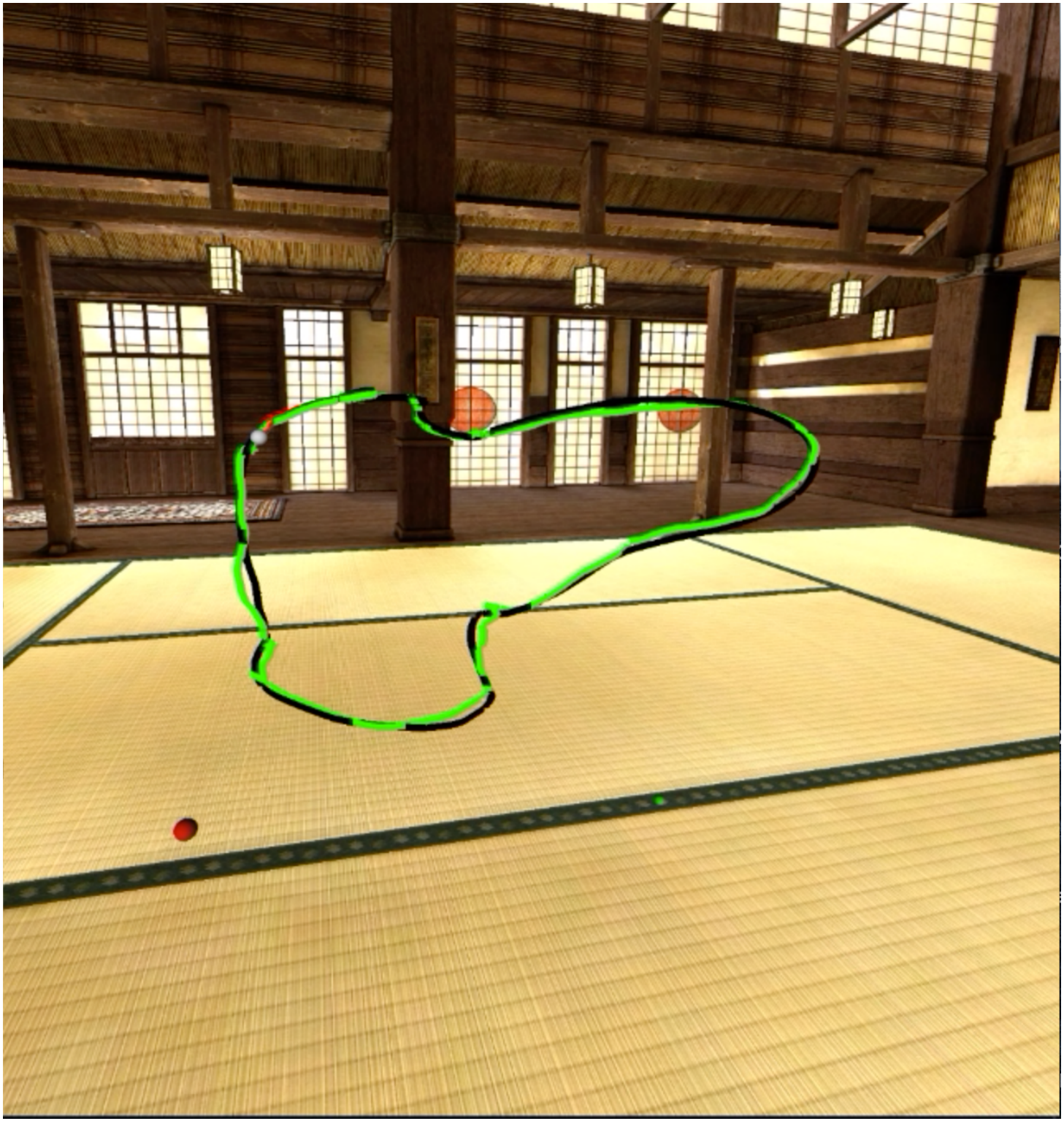
l=0.5

Xs are relatively uncorrelated

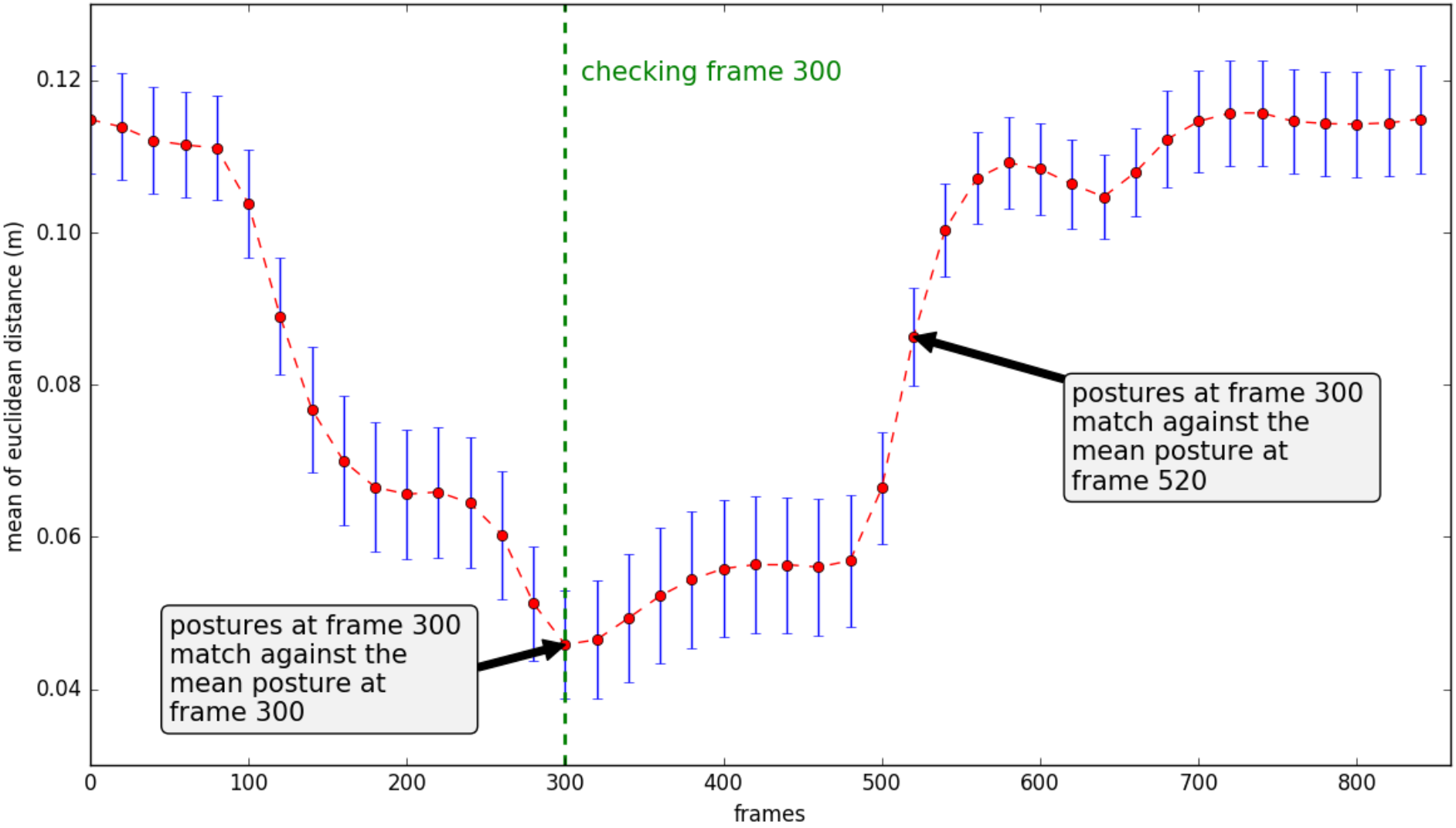
l=2.0

Xs are are more uncorrelated


03
02
00
01
07
19
08
31
20
09
30
21
14
10
11
15
16
26
27
24
28
12
13
17
36
43
23
29
25
24
37
44
38
45
39
46
40
42
49
47
THE MUSCULAR SYSTEM



The postures at frame 300 do not kook like postures at any other frame



Dynamics: Notation

Symbol	Meaning
$\boldsymbol{x}$	position or state of one or more rigid bodies
$\dot{\boldsymbol{x}}$	velocity (usu. linear and angular)
$\ddot{\boldsymbol{x}}$	acceleration (usu. linear and angular)
$\boldsymbol{R}$	rotation matrix representing orientation of a body
$\boldsymbol{\omega}$	angular velocity
$\boldsymbol{q}$	quaternion representation of an orientation or rotation
m	mass of a single rigid body
$\boldsymbol{M}$	mass matrix
$\boldsymbol{I}$	identity matrix
$\boldsymbol{\mathcal{I}}$	moment of inertia tensor
n_b, n_c	number of bodies, number of constraints
$\boldsymbol{\alpha}, \boldsymbol{\beta}$	stabilizing parameters added to the equations of motion
$\phi()$	error or energy function for a single constraint
$\boldsymbol{J}$	matrix of partial derivatives of constraint error functions
h	timestep
$\boldsymbol{f}$	forces (and torques)
$\boldsymbol{\tau}$	torques
$\boldsymbol{\lambda}$	constraint forces

Table 1: Meanings of specific symbols used to discuss dynamic simulation

Newton's Laws

$$\begin{bmatrix} \boldsymbol{f}_{1t} \\ \boldsymbol{\tau}_{1t} \\ \boldsymbol{f}_{2t} \\ \boldsymbol{\tau}_{2t} \end{bmatrix} = \begin{bmatrix} m_1\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}}_{1t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & m_2\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{I}}_{2t} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{1t} \\ \dot{\boldsymbol{\omega}}_{1t} \\ \ddot{\boldsymbol{x}}_{2t} \\ \dot{\boldsymbol{\omega}}_{2t} \end{bmatrix} \Rightarrow \boldsymbol{f}_t = \boldsymbol{M}_t\ddot{\boldsymbol{x}}_t$$

Forces have **gyroscopic**, **gravity** and **control** components

$$\boldsymbol{f} = \boldsymbol{f_c} + \boldsymbol{f_g} + \boldsymbol{f_u}.$$

Semi-implicit Euler uses future velocity for computing position

$$\dot{\boldsymbol{x}}_{t+h} = \dot{\boldsymbol{x}}_t + h\boldsymbol{M}^{-1}\boldsymbol{f}_t \tag{1}$$

$$\boldsymbol{x}_{t+h} = \boldsymbol{x}_t + h\dot{\boldsymbol{x}}_{t+h} \tag{2}$$

Handling Joint Constraints

$$\boldsymbol{J}_t = \nabla \boldsymbol{\phi}(\boldsymbol{x}_t) = \begin{bmatrix} \frac{\partial \phi_1}{\partial x_{1t}} & \cdots & \frac{\partial \phi_1}{\partial x_{(6n_b)t}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_k}{\partial x_{1t}} & \cdots & \frac{\partial \phi_k}{\partial x_{(6n_b)t}} \end{bmatrix}$$

Reduce joint constraints gradually during the integration

$$(I - \boldsymbol{\alpha})\boldsymbol{\phi}(\boldsymbol{x}_t) = \boldsymbol{\phi}(\boldsymbol{x}_{t+h}) \approx \boldsymbol{\phi}(\boldsymbol{x}_t) + \boldsymbol{J}_t(\boldsymbol{x}_{t+h} - \boldsymbol{x}_t)$$

$$\dot{\boldsymbol{x}}_{t+h} = \dot{\boldsymbol{x}}_t + h\boldsymbol{M}^{-1}\boldsymbol{f}_t \tag{1}$$

$$\boldsymbol{x}_{t+h} = \boldsymbol{x}_t + h\dot{\boldsymbol{x}}_{t+h} \tag{2}$$

$$(I - \boldsymbol{\alpha})\boldsymbol{\phi}(\boldsymbol{x}_t) = \boldsymbol{\phi}(\boldsymbol{x}_{t+h}) \approx \boldsymbol{\phi}(\boldsymbol{x}_t) + \boldsymbol{J}_t(\boldsymbol{x}_{t+h} - \boldsymbol{x}_t) \tag{3}$$

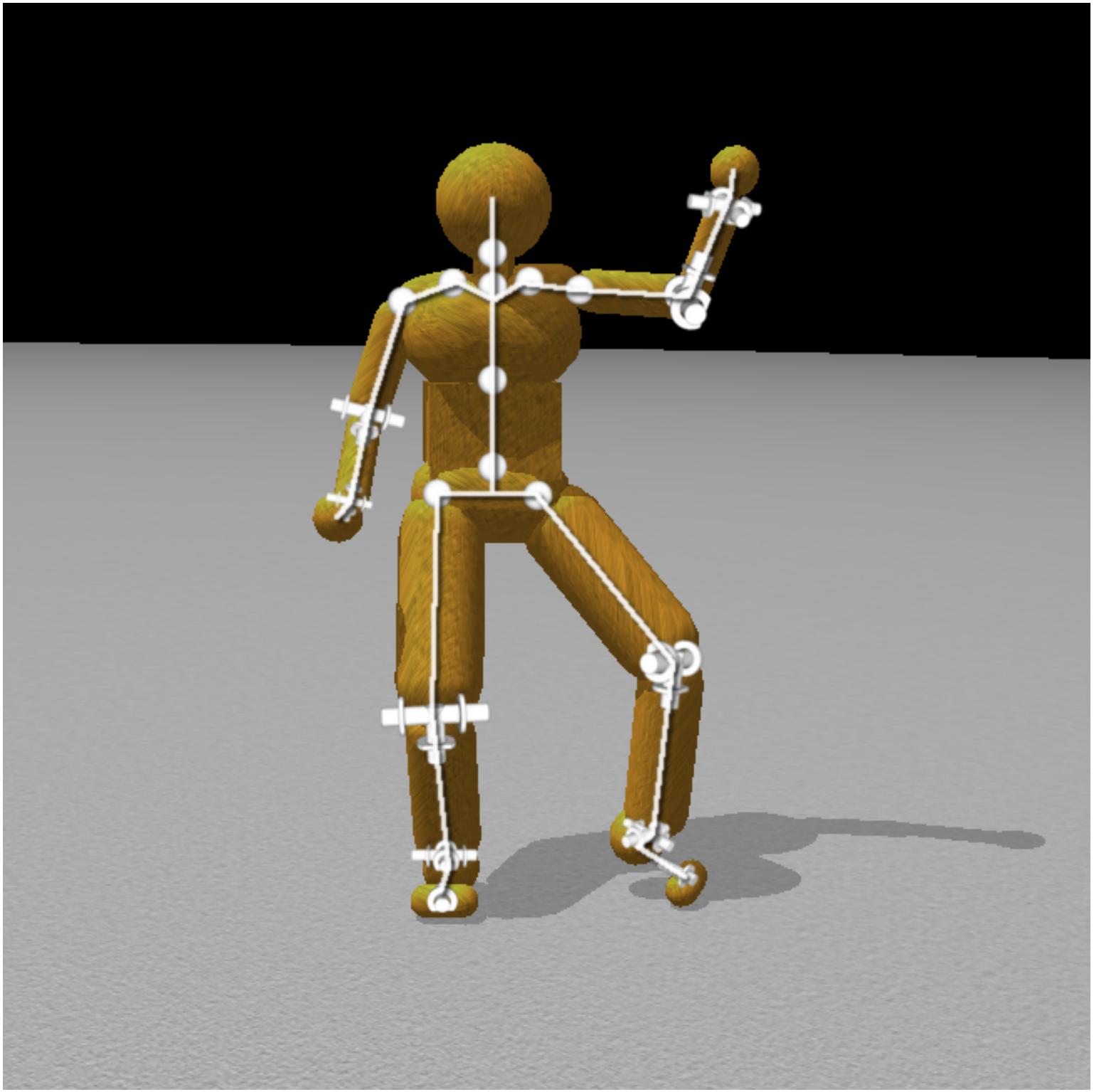
$$\boldsymbol{x}_{t+h} = \boldsymbol{x}_t + h\dot{\boldsymbol{x}}_t + h^2\boldsymbol{M}_t^{-1}\left(\boldsymbol{f}_{ct} + \boldsymbol{f}_{gt} + \boldsymbol{f}_{ut}\right) \tag{4}$$

Use 1~4 to eliminate all future quantities t+h

$$(I - \boldsymbol{\alpha})\boldsymbol{\phi}(\boldsymbol{x}_t) = \boldsymbol{\phi}(\boldsymbol{x}_t) + \boldsymbol{J}_t\left(\boldsymbol{x}_t + h\dot{\boldsymbol{x}}_t + h^2\boldsymbol{M}_t^{-1}\left(\boldsymbol{f}_{ct} + \boldsymbol{f}_{gt} + \boldsymbol{f}_{ut}\right) - \boldsymbol{x}_t\right)$$

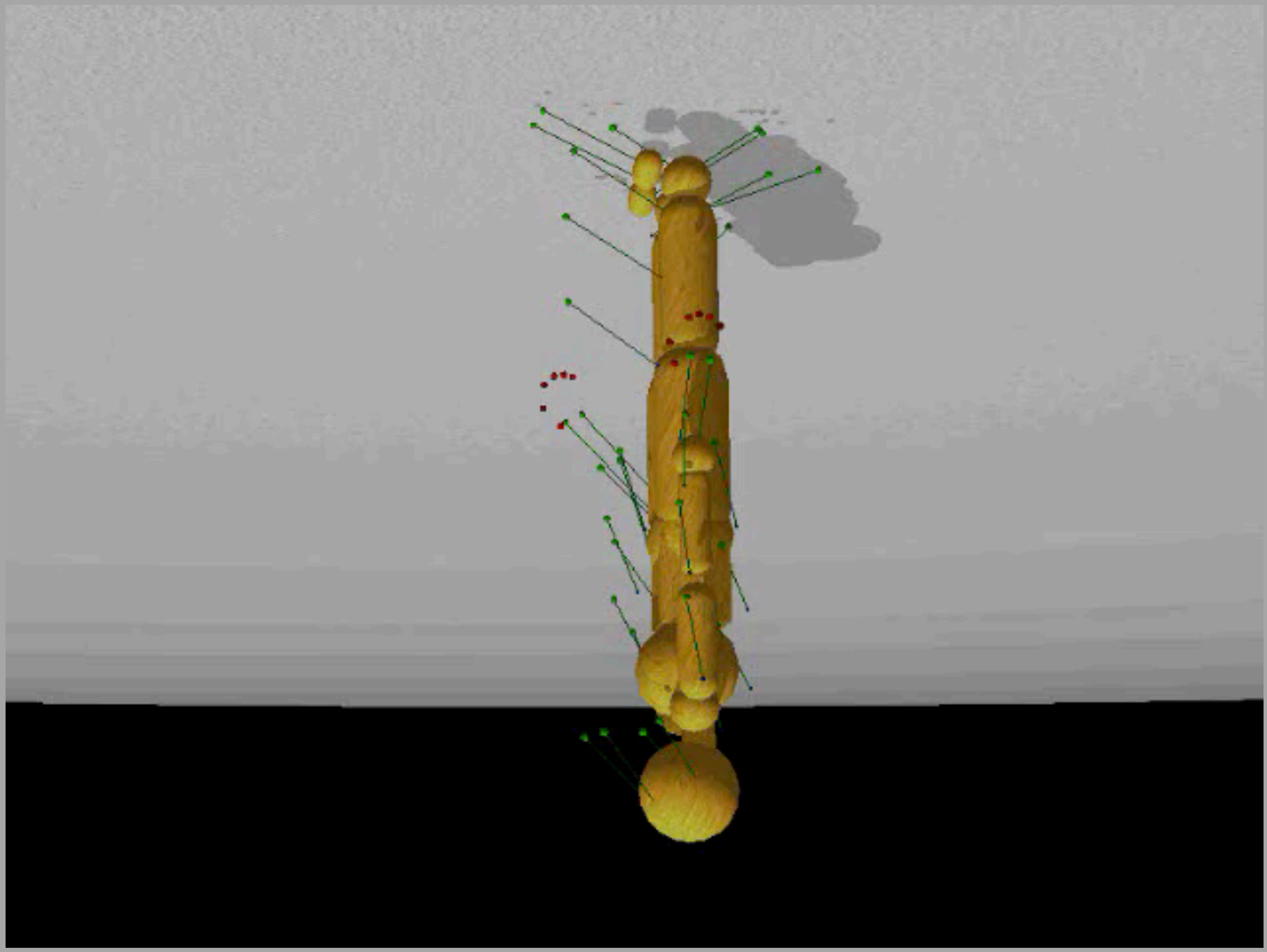
Solve for $\boldsymbol{f}_{ct}$

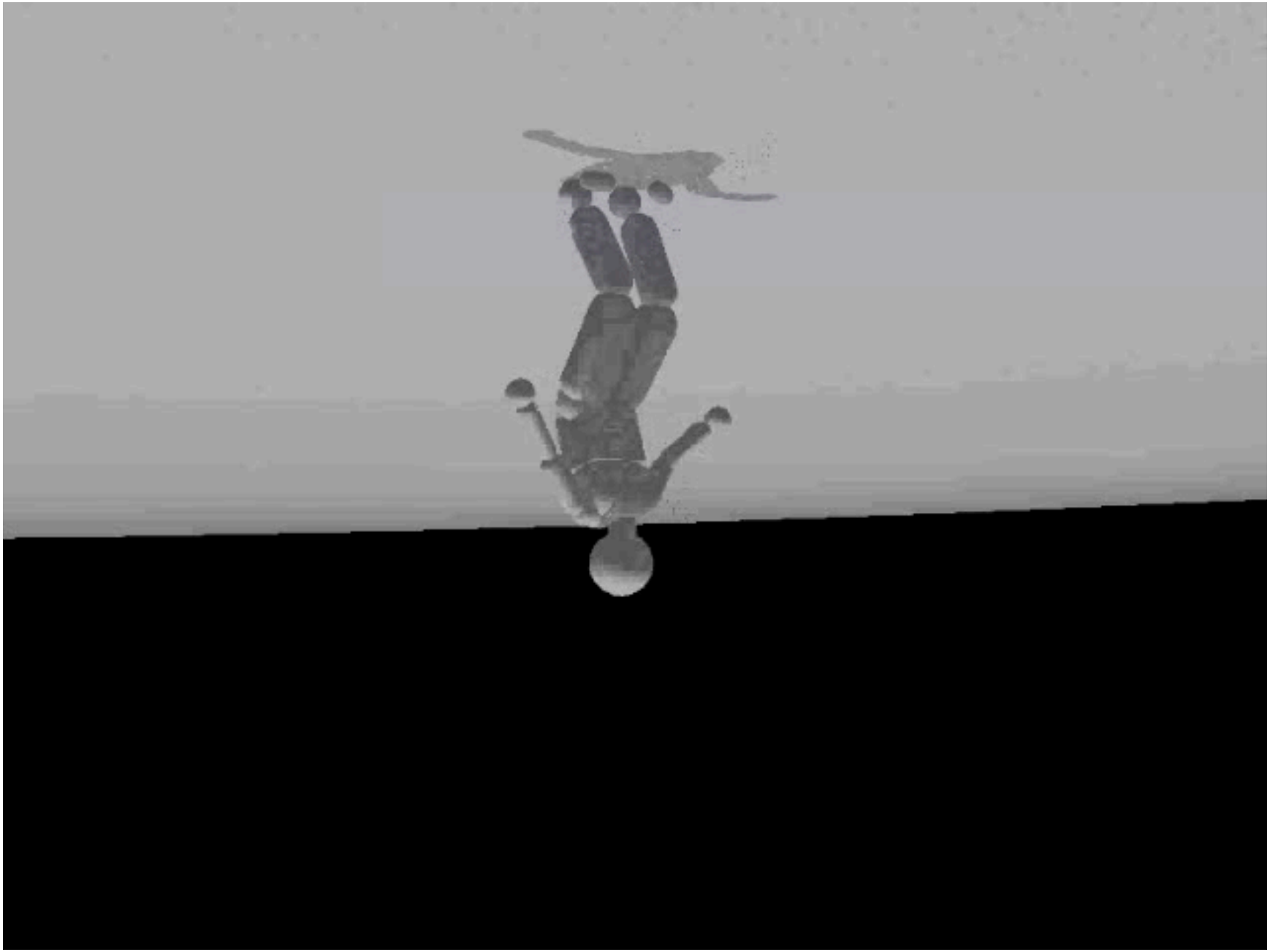
$$\boldsymbol{J}_t\boldsymbol{M}_t^{-1}\boldsymbol{f}_{ct} = -\frac{1}{h^2}\boldsymbol{\alpha}\boldsymbol{\phi}(\boldsymbol{x}_t) - \frac{1}{h}\boldsymbol{J}_t\dot{\boldsymbol{x}}_t - \boldsymbol{J}_t\boldsymbol{M}_t^{-1}\left(\boldsymbol{f}_{gt} + \boldsymbol{f}_{ut}\right)$$

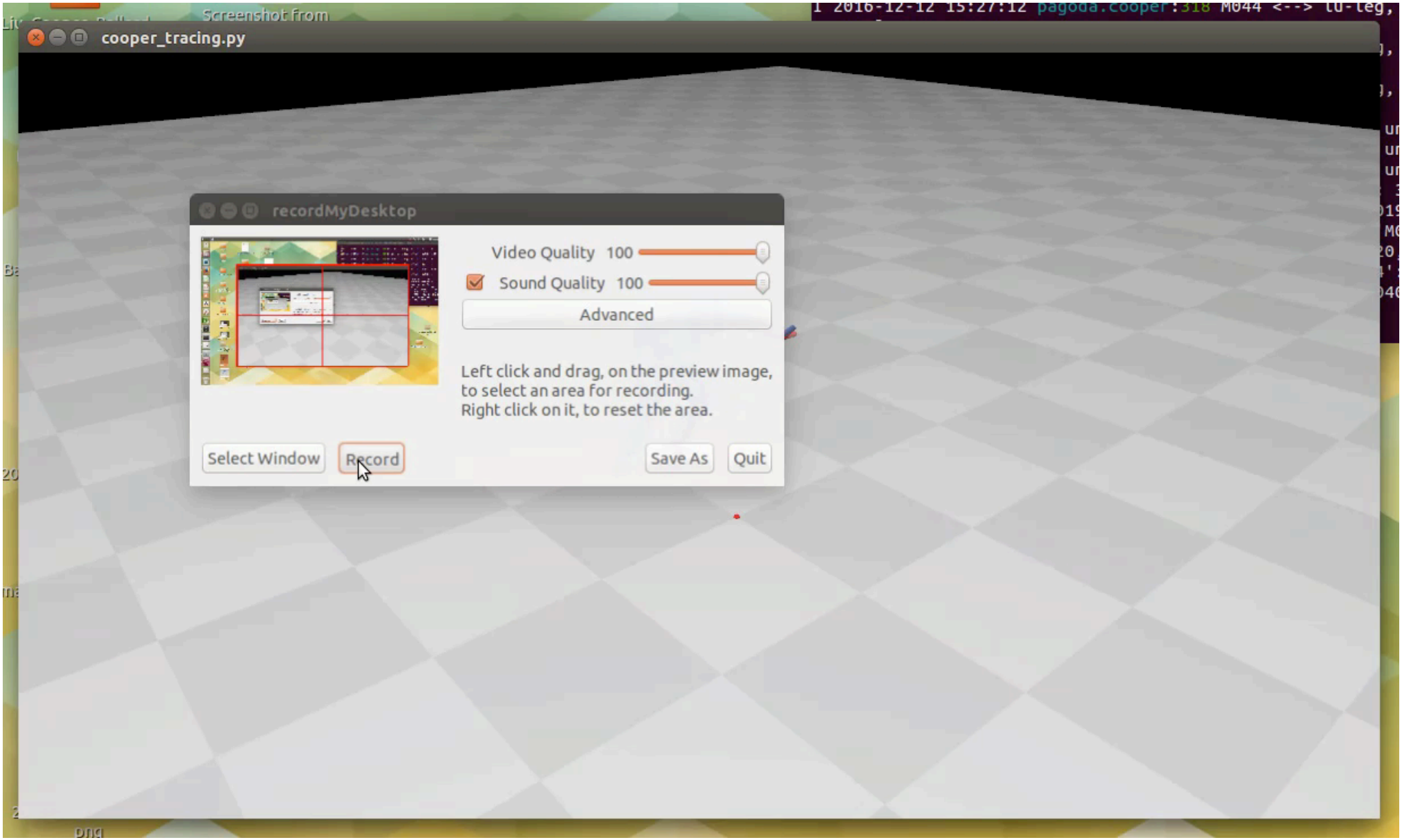


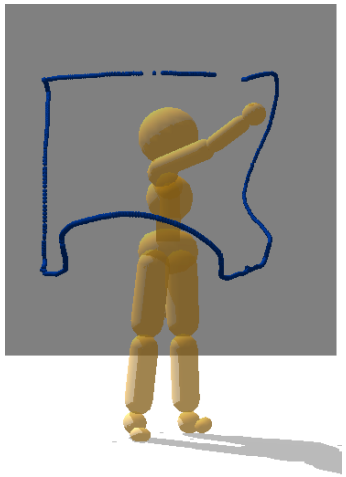
Dynamic Model

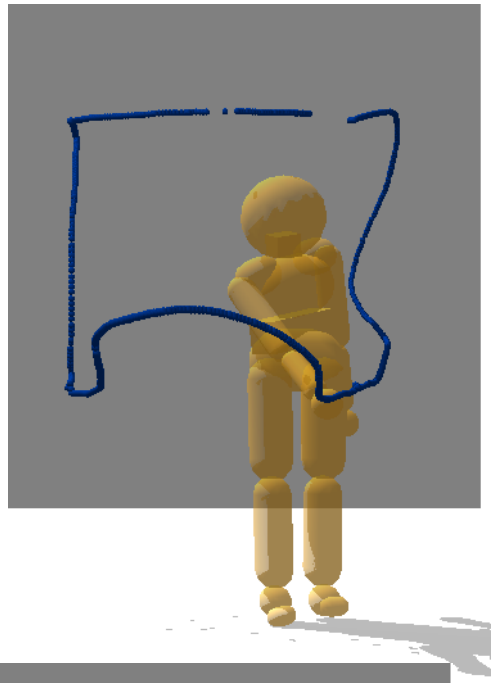
Including
50 DOFs,
Lengths,
Masses,
Inertial Tensors

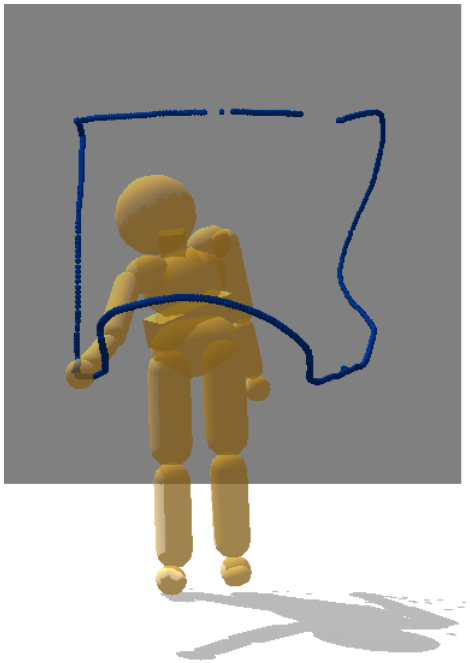


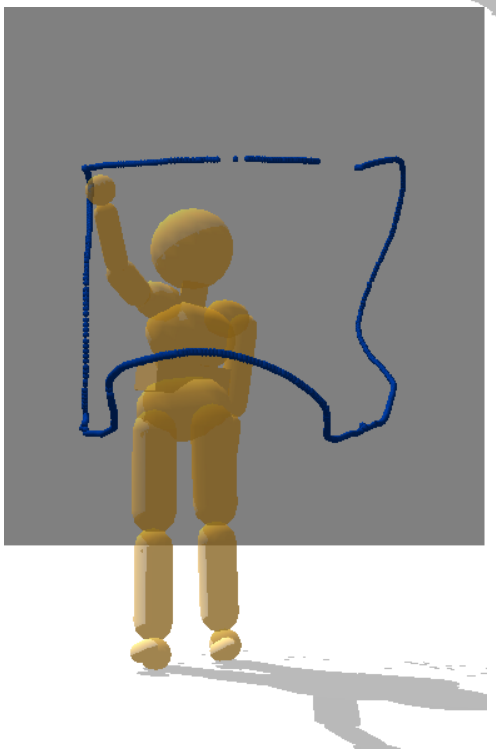



Screenshot from
cooper_tracing.py
recordMyDesktop
Video Quality 100
Sound Quality 100
Advanced
Left click and drag, on the preview image,
to select an area for recording.
Right click on it, to reset the area.
Select Window
Record
Save As
Quit

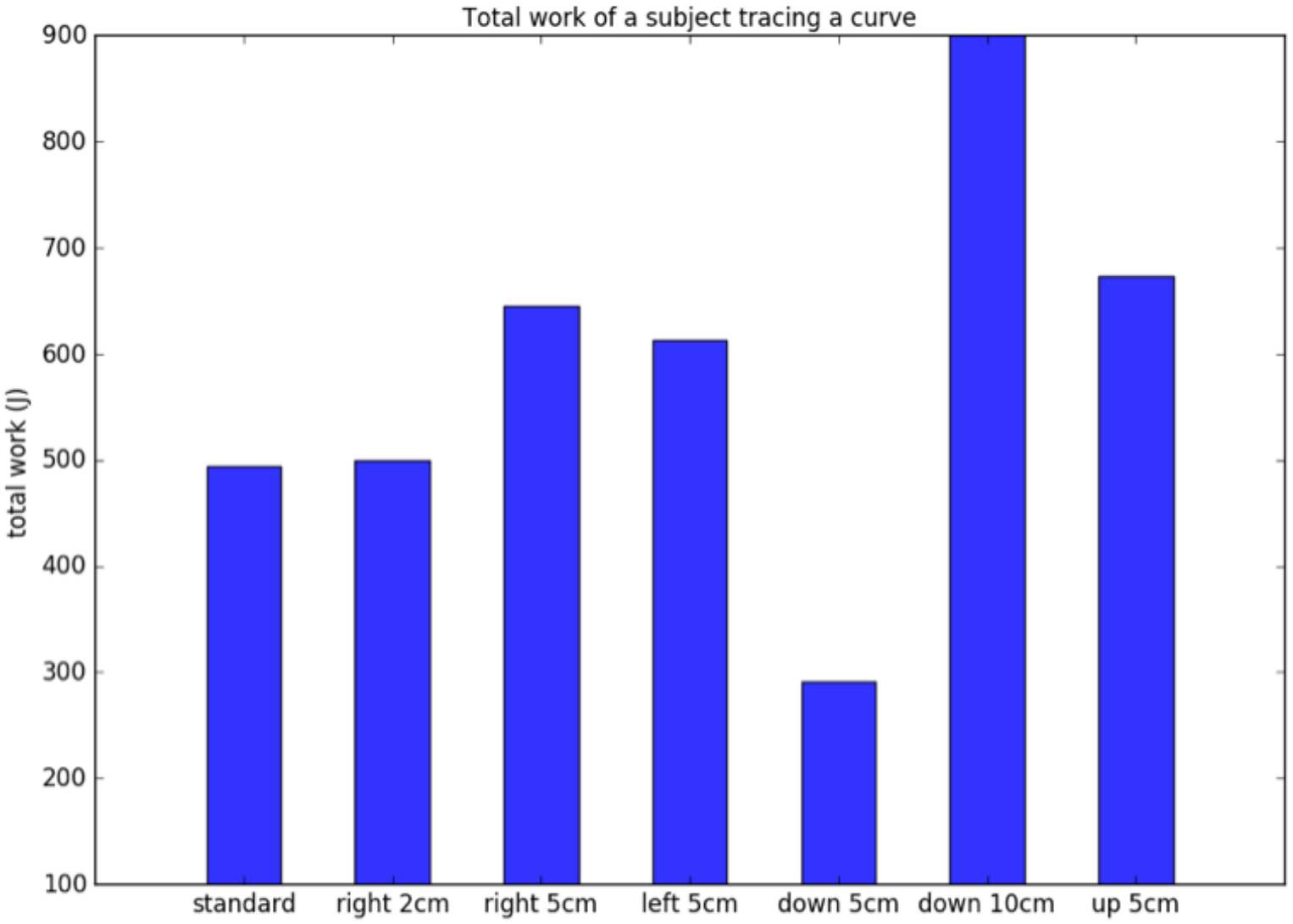


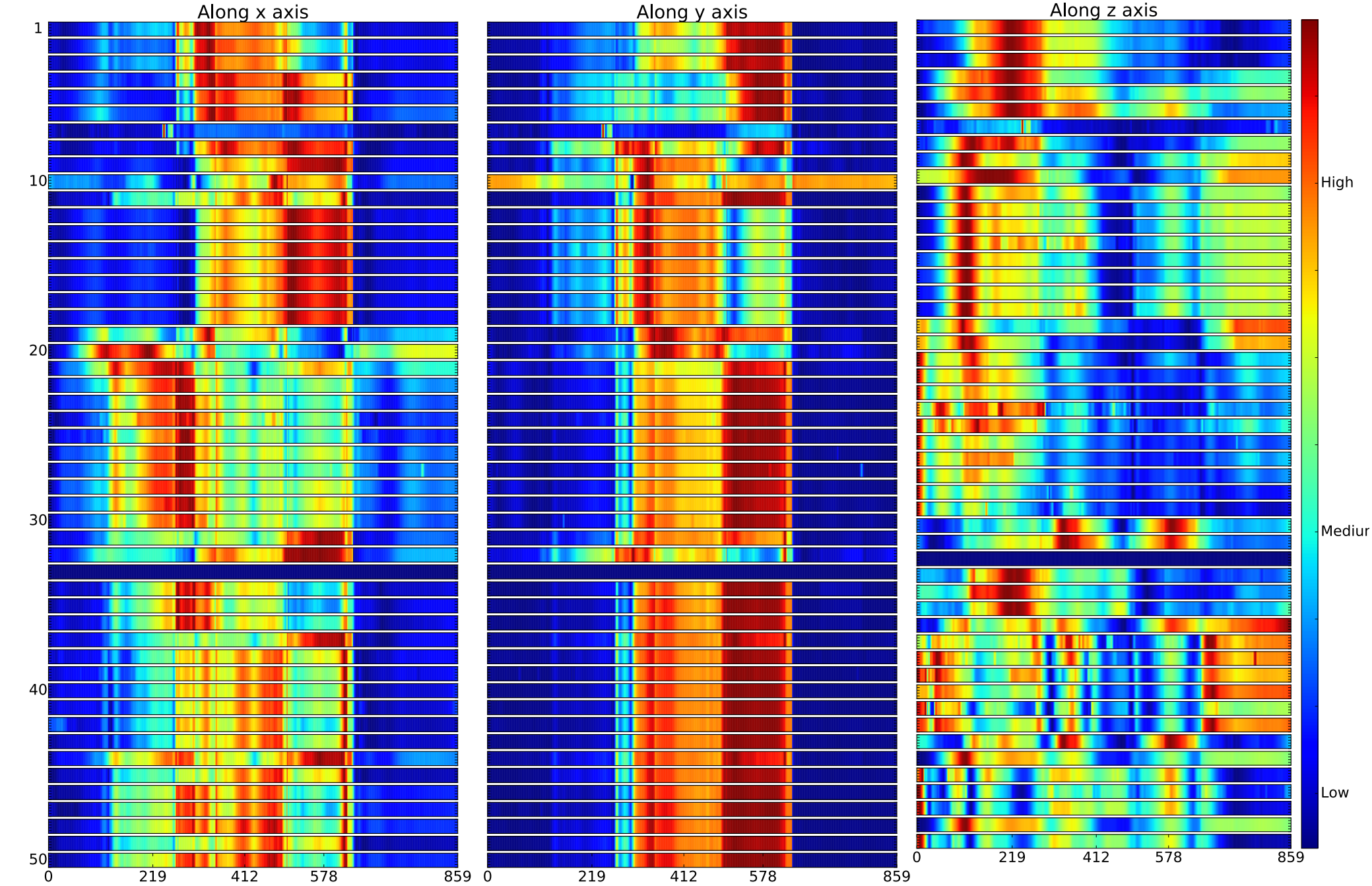






One subject traces the square curve




Along x axis
Along y axis
Along z axis
1
10
20
30
40
50
0
219
412
578
859
High
Mediur
Low