Robots that Learn

*Machine Learning for smarter and efficient actuation: **Lectures 2&3***

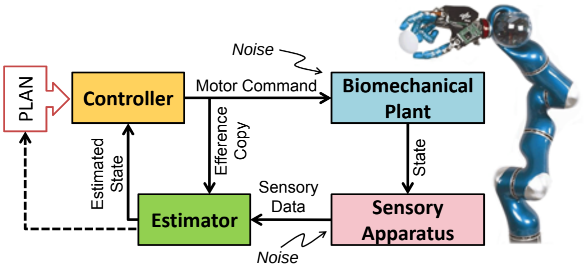
Professor Sethu Vijayakumar FRSE
Microsoft Research RAEng Chair in Robotics
University of Edinburgh, UK
http://homepages.inf.ed.ac.uk/svijayak

Director, **Edinburgh Centre for Robotics**
www.edinburgh-robotics.org

- Ask Questions!
- Disclaimer: It is impossible to cover ALL **machine learning** techniques for the large variety of **robotics** problems...

... we will largely ignore the sensing issues!
.... we will focus on non-parametric methods.

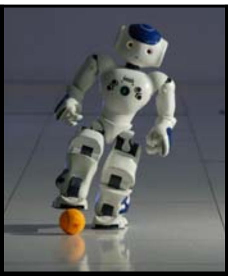
Robot Sensorimotor Control



Robots: Sense, Plan, Move

Interesting **Machine Learning Challenges** in each domain

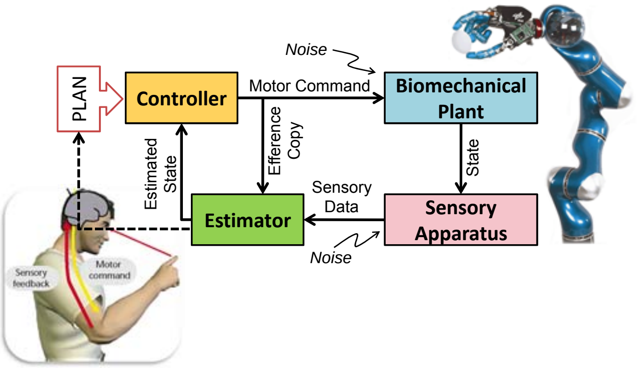
- **Sensing**
  - Incomplete state information, Noise
  - Unknown causal structure
- **Planning**
  - Optimal Redundancy resolution
  - Incomplete knowledge of appropriate optimization cost function
- **Moving**
  - Incomplete knowledge of (hard to model) nonlinear dynamics
  - Dynamically changing motor functions: wear and tear/loads
- **Representation**
  - Uncovering suitable representation



Autonomous
Make decisions on its own
Adapt to changing world
React to unseen scenarios

Autonomous?
Make decisions on its own: Game Playing Engine
~~Adapt to changing world~~
~~React to unseen scenarios~~

Sensorimotor Control



Optimal Control

Given:

- Start state,
- fixed-time horizon T and
- system dynamics $dx = f(x,u)dt + F(x,u)d\omega$

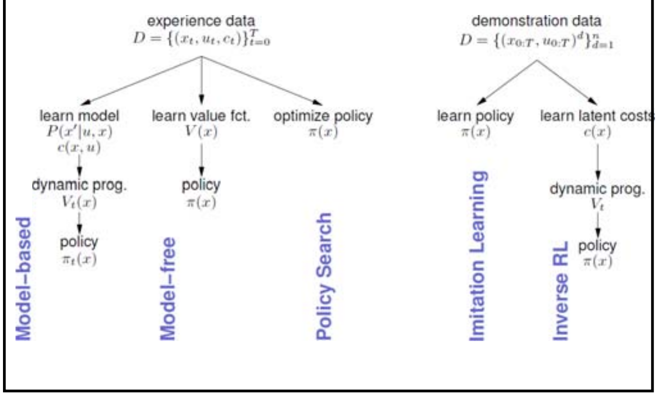
How the system reacts (Δx) to forces (u)

And assuming some cost function:

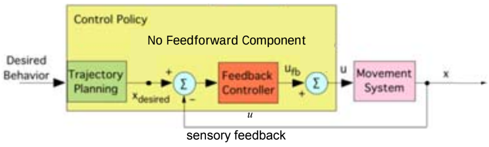
$$v^{\pi}(t,x) \equiv E\left[\underbrace{h(x(T))}_{\text{Final Cost}} + \underbrace{\int_t^T l(\tau, x(\tau), \pi(\tau, x(\tau)))d\tau}_{\text{Running Cost}} \right]$$

Apply **Statistical Optimization** techniques to find optimal **control commands**

Aim: find control law π* that minimizes $v^{\pi}(0, x_0)$.



Feedforward Predictive Control

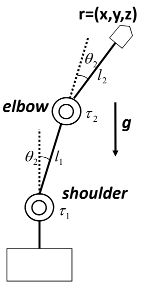


$$u_{fb} = k_p(x_{cur} - x_{des}) + k_d(\dot{x}_{cur} - \dot{x}_{des})$$

What is inside a feedforward controller?

Typically a (predictive) model of the robot.

Dynamics Model: 2DOF arm

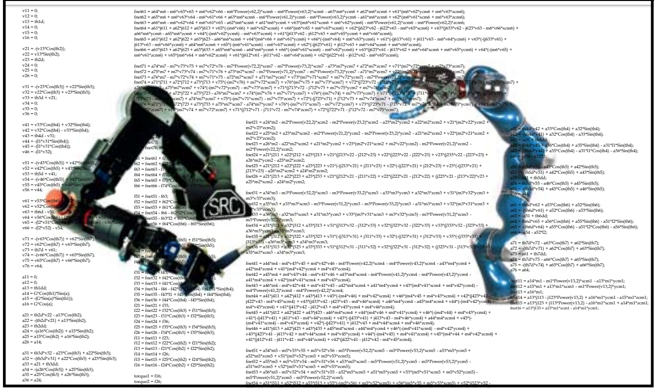


$$\left.\begin{aligned}
\tau_1 &= (I_1 + m_1 L_{G1}^2 + I_2 + m_2 L_{G2}^2 + m_2 L_1^2)\ddot{\theta}_1 \\
&\quad + (I_2 + m_2 L_{G2}^2)\ddot{\theta}_2 \\
&\quad + (2 m_2 L_1 L_{G2}^2 \ddot{\theta}_2 + m_2 L_1 L_{G2}^2 \ddot{\theta}_1)\cos\theta_2 \\
&\quad - (m_1 L_1 L_{G2} \ddot{\theta}_2{}^2 + m_1 L_1 L_{G2} \dot{\theta}_1 \dot{\theta}_2)\sin\theta_2 \\
&\quad - (m_1 L_{G1} + m_2 L_1) g \sin\theta_1 \\
&\quad - m_2 L_{G1} g \sin\theta_1 \cos\theta_2 \\
&\quad - m_2 L_{G2} g \sin\theta_2 \cos\theta_1 \\
\tau_2 &= (I_2 + m_2 L_{G2}^2)\ddot{\theta}_1 + (I_2 + m_2 L_{G2}^2)\ddot{\theta}_2 \\
&\quad + m_2 L_1 L_{G2} \ddot{\theta}_1 \cos\theta_2 \\
&\quad + m_2 L_1 L_{G2} \dot{\theta}_2 \sin\theta_2 \\
&\quad - m_2 L_{G2} g \sin\theta_1 \cos\theta_2 \\
&\quad - m_2 L_{G2} g \sin\theta_2 \cos\theta_1
\end{aligned}\right\}$$

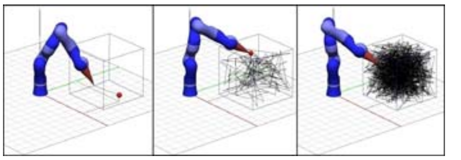
Inverse Dynamics derived based on ***equations of motion*** *and* ***rigid body dynamics*** *assumptions*

Can be used as a feedforward model and minor deviations corrected through feedback control

Robot Dynamics: 7DOF

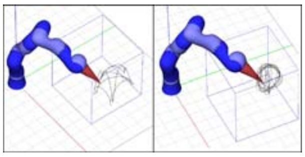


Data from Motor Babbling

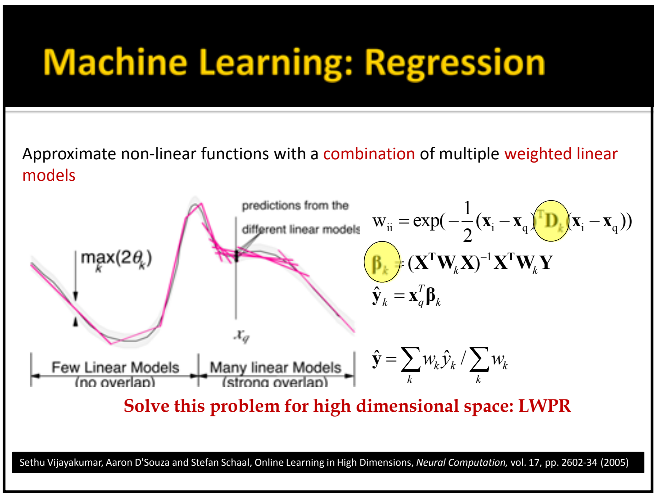


Random motions in a specified work area

$$\tau = f(\theta, \dot{\theta}, \ddot{\theta})$$



Kinesthetic demo using a dynamic target



Machine Learning: Regression

Approximate non-linear functions with a combination of multiple weighted linear models

$$w_{ii} = \exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_q)^T \mathbf{D}_k (\mathbf{x}_i - \mathbf{x}_q))$$

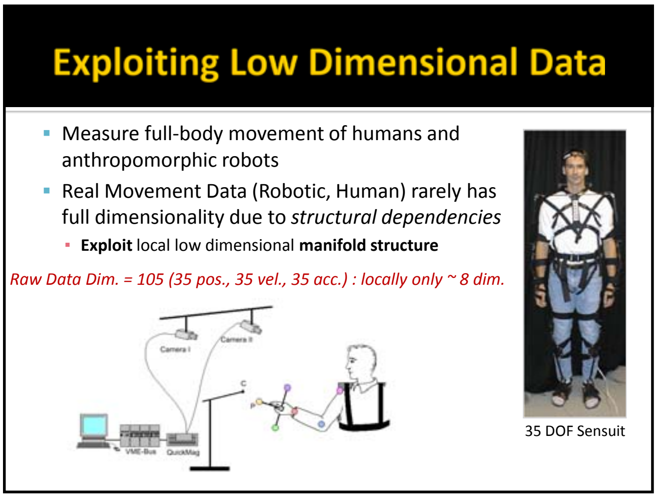
$$\boldsymbol{\beta}_k = (\mathbf{X}^T\mathbf{W}_k\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}_k\mathbf{Y}$$

$$\hat{\mathbf{y}}_k = \mathbf{x}_q^T\boldsymbol{\beta}_k$$

$$\hat{\mathbf{y}} = \sum_k w_k \hat{y}_k / \sum_k w_k$$

Solve this problem for high dimensional space: LWPR

Sethu Vijayakumar, Aaron D'Souza and Stefan Schaal, Online Learning in High Dimensions, *Neural Computation*, vol. 17, pp. 2602-34 (2005)

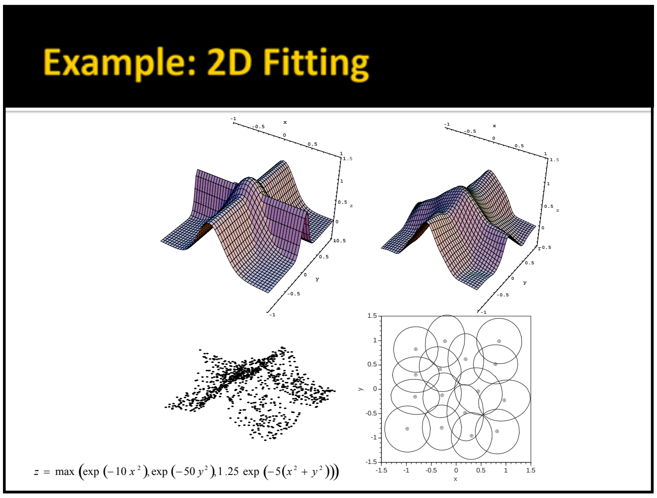


Exploiting Low Dimensional Data

- Measure full-body movement of humans and anthropomorphic robots
- Real Movement Data (Robotic, Human) rarely has full dimensionality due to *structural dependencies*
  - **Exploit** local low dimensional **manifold structure**

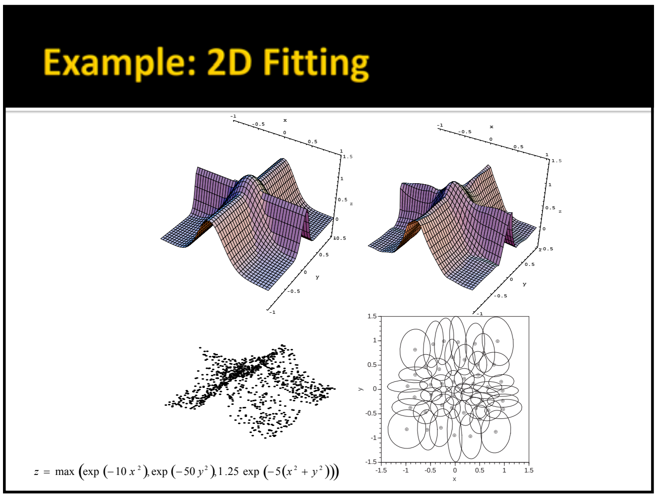
Raw Data Dim. = 105 (35 pos., 35 vel., 35 acc.) : locally only ~ 8 dim.

35 DOF Sensuit



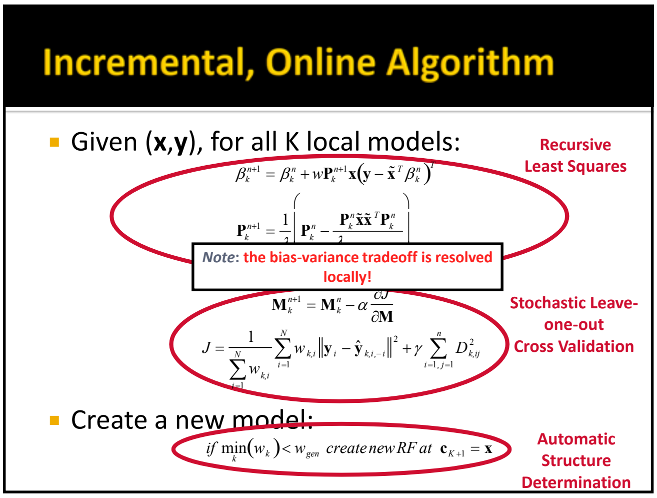
Example: 2D Fitting

$z = \max(\exp(-10x^2), \exp(-50y^2), 1.25\exp(-5(x^2+y^2)))$



Example: 2D Fitting

$z = \max(\exp(-10x^2), \exp(-50y^2), 1.25\exp(-5(x^2+y^2)))$



Incremental, Online Algorithm

- Given (**x**,**y**), for all K local models:

$$\beta_k^{n+1} = \beta_k^n + w\mathbf{P}_k^{n+1}\mathbf{x}(\mathbf{y} - \tilde{\mathbf{x}}^T\beta_k^n)^T$$

$$\mathbf{P}_k^{n+1} = \frac{1}{\lambda}\left(\mathbf{P}_k^n - \frac{\mathbf{P}_k^n\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\mathbf{P}_k^n}{\lambda}\right)$$

Recursive Least Squares

***Note*: the bias-variance tradeoff is resolved locally!**

$$\mathbf{M}_k^{n+1} = \mathbf{M}_k^n - \alpha\frac{\partial J}{\partial \mathbf{M}}$$

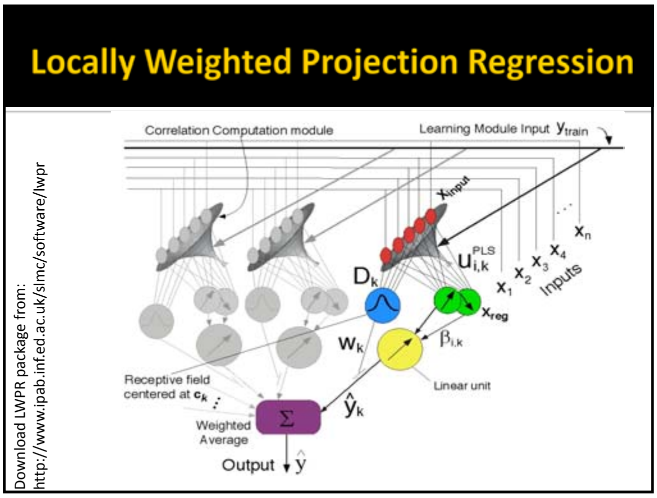
$$J = \frac{1}{\sum_{i=1}^N w_{k,i}}\sum_{i=1}^N w_{k,i}\|\mathbf{y}_i - \hat{\mathbf{y}}_{k,i,-i}\|^2 + \gamma\sum_{i=1,j=1}^n D_{k,ij}^2$$

Stochastic Leave-one-out Cross Validation

- Create a new model:

$$if\ \min_k(w_k) < w_{gen}\ create\ new\ RF\ at\ \mathbf{c}_{K+1} = \mathbf{x}$$

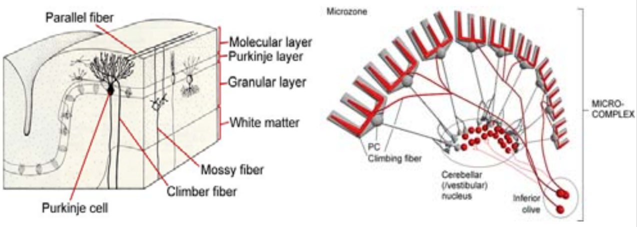
Automatic Structure Determination

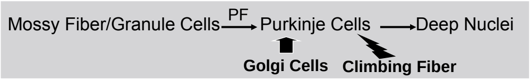


Locally Weighted Projection Regression

Download LWPR package from: http://www.ipab.inf.ed.ac.uk/slmc/software/lwpr

Cerebellum:
A seat for internal models

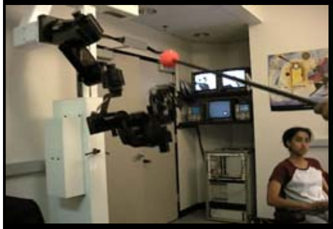



Mossy Fiber/Granule Cells $\xrightarrow{\text{PF}}$ Purkinje Cells $\longrightarrow$ Deep Nuclei

Golgi Cells **Climbing Fiber**

Online Learning with LWPR

Learning the Internal Dynamics



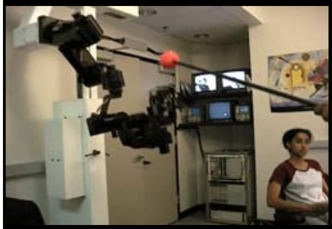
Learning the Task Dynamics

Stefan Klanke, Sethu Vijayakumar and Stefan Schaal, A Library for Locally Weighted Projection Regression, *Journal of Machine Learning Research (JMLR)*, vol. 9. pp. 623--626 (2008).

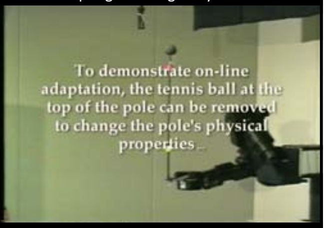
http://www.ipab.inf.ed.ac.uk/slmc/software/lwpr

Online Learning with LWPR

Learning the Internal Dynamics



Adapting to Changed Dynamics



Stefan Klanke, Sethu Vijayakumar and Stefan Schaal, A Library for Locally Weighted Projection Regression, *Journal of Machine Learning Research (JMLR)*, vol. 9. pp. 623--626 (2008).

http://www.ipab.inf.ed.ac.uk/slmc/software/lwpr

Using Dynamics in the Control Loop

Plant $\dot{x} = f(x) + u$

Problem

Given a desired trajectory $x_d(t)$,
Design a control law $u = u(x, \dot{x}, x_d, \dot{x}_d)$

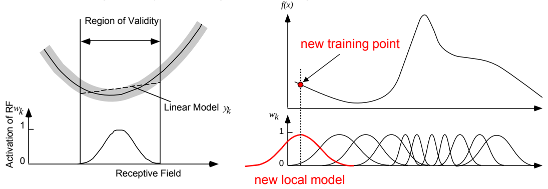
which can achieve asymptotic tracking
$e = x - x_d \to 0$ as $t \to \infty$

Known system dynamics feedback linearization + PD
- control law $u = -f(x) + \dot{x}_d - K(x - x_d)$
- tracking error dynamics $\dot{e} = -Ke$ where $e = x - x_d$

Incremental Robust Function Approx.

Locally Weighted Regression

- Approximate nonlinear function by locally linear models
- Create new local models as needed
- Update the linear model in each kernel by weighted recursive least squares
- Adjust the size and shape of receptive field by gradient descent
- Estimation is given by the weighted average of all local models



Stefan Klanke, Sethu Vijayakumar and Stefan Schaal, A Library for Locally Weighted Projection Regression, *Journal of Machine Learning Research (JMLR)*, vol. 9. pp. 623--626 (2008).

Nonlinear learning of plant dynamics

Plant $\dot{x} = f(x) + u$

Locally linear approximation

Locally linear models

$$y_k(x) = \bar{\mathbf{x}}_k^T \boldsymbol{\theta}_k,\ \bar{\mathbf{x}}_k = \begin{bmatrix} x - c_k \\ 1 \end{bmatrix},\ \boldsymbol{\theta}_k = \begin{bmatrix} b_k \\ b_{0,k} \end{bmatrix}$$

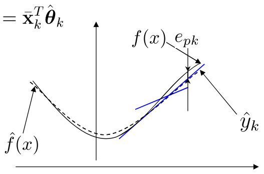
Estimate of the function

$$\hat{f}(x) = \frac{\sum_{k=1}^{K} w_k(x) \hat{y}_k(x)}{\sum_{k=1}^{K} w_k(x)} \quad \hat{y}_k(x) = \bar{\mathbf{x}}_k^T \hat{\boldsymbol{\theta}}_k$$

Error sources

-Tracking error $e = x - x_d$

-Estimation error of each local model
$e_{pk} = f(x) - \hat{y}_k$



Composite Adaptation

Composite adaptation

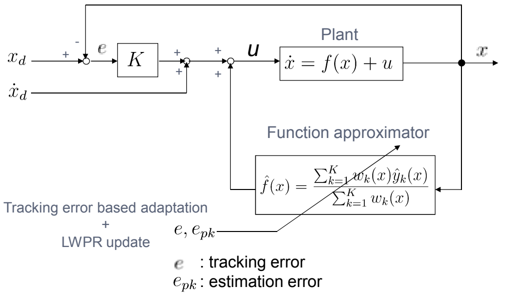
- parameter update by tracking error+estimation error
  - tracking error $e = x - x_d$ [Slotine et al.]
  - estimation error $e_{pk} = y - \hat{y}_k$
- tracking error based adaptation + LWPR update

$$\dot{\hat{\boldsymbol{\theta}}}_k = \mathbf{P}_k \bar{\mathbf{x}}_k \underline{(\bar{w}_k e + w_k e_{pk})}$$
$$\dot{\mathbf{P}}_k = \lambda \mathbf{P}_k - w_k \mathbf{P}_k \bar{\mathbf{x}}_k \bar{\mathbf{x}}_k^T \mathbf{P}_k$$

driven by tracking error + estimation error

- closed-loop stability can be shown by Lyapunov analysis

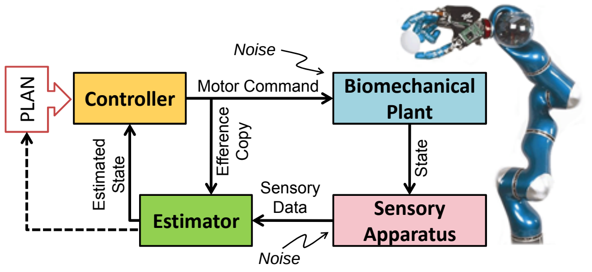
The Composite Controller



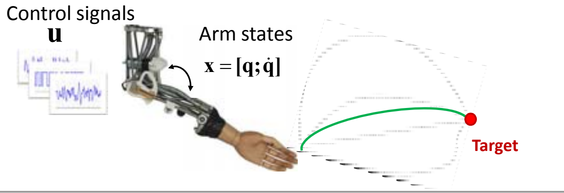
e : tracking error
e_{pk} : estimation error

Elements needed for computing control signal: $u = -\hat{f}(x) + \dot{x}_d - K(x - x_d)$

Sensorimotor Control



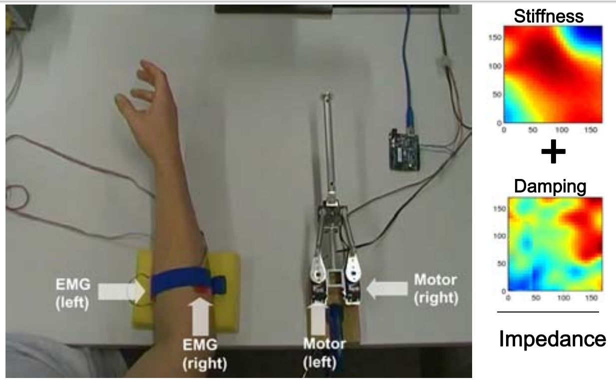
Planning with Redundancy



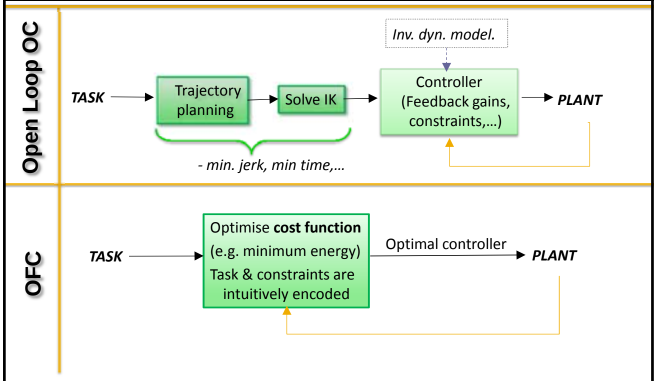
Redundancy at various levels:

- Task -> End Effector Trajectory *(Min. Jerk, Min. Energy etc.)*
- End Effector -> Joint Angles *(Inverse Kinematics)*
- Joint Angles -> Joint Torques *(Inverse Dynamics)*
- Joint Torques -> Joint Stiffness *(Variable Impedance)*

Variable Stiffness Actuation



Plan Optimization and Control



Optimal Feedback Control

Given:
- Start & end states,
- fixed-time horizon T and
- system dynamics $dx = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\omega$

How the system reacts (Δx) to forces (u)

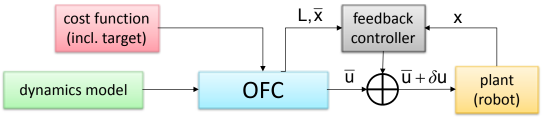
And assuming some cost function:

$$v^{\pi}(t,\mathbf{x}) \equiv E\left[\underbrace{h(\mathbf{x}(T))}_{\text{Final Cost}} + \underbrace{\int_t^T l(\tau,\mathbf{x}(\tau),\pi(\tau,\mathbf{x}(\tau)))d\tau}_{\text{Running Cost}}\right]$$

Apply **Statistical Optimization** techniques to find optimal **control commands**

Aim: find control law π* that minimizes $v^{\pi}(0, x_0)$.

What does an OFC generate?



OFC law

$$\boxed{\begin{aligned} \mathbf{u}_k^{plant} &= \bar{\mathbf{u}}_k + \delta\mathbf{u}_k \\ \delta\mathbf{u}_k &= \mathbf{L}_k \cdot (\mathbf{x}_k - \bar{\mathbf{x}}_k) \end{aligned}}$$

Choice of Optimization Methods

- Analytic Methods
  - Linear Quadratic Regulator (LQR)
  - Linear Quadratic Gaussian (LQG)
- Local Iterative Methods
  - iLQG, iLDP
- Dynamic Programming (DDP)
- Inference based methods
  - AICO, PI^2, ψ-Learning

Variable Impedance Policies
-- through Stochastic Optimization

Assume knowledge of **actuator dynamics**
Assume knowledge of **cost** being optimized

- Explosive Movement Tasks (e.g., throwing)
- Periodic Movement Tasks and Temporal Optimization (e.g. walking, brachiation)
- Learning dynamics (OFC-LD)

Optimal Variable Impedance

Assume knowledge of **actuator dynamics**
Assume knowledge of **cost** being optimized

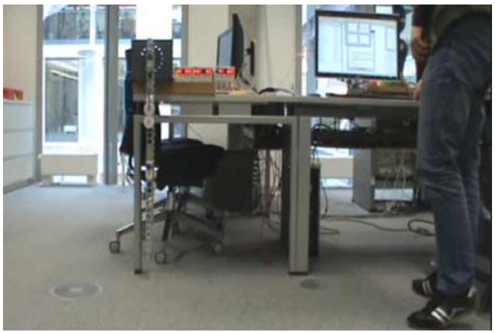
- Explosive Movement Tasks (e.g., throwing)
- Periodic Movement Tasks and Temporal Optimization (e.g. walking, brachiation)
- Learning dynamics (OFC-LD)

David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)

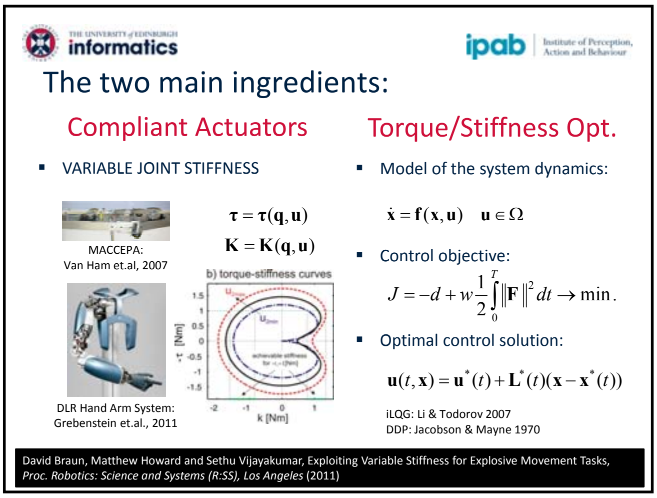
THE UNIVERSITY of EDINBURGH informatics

ipab Institute of Perception, Action and Behaviour

Highly **dynamic** tasks, explosive movements



David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)



The two main ingredients:

Compliant Actuators

- VARIABLE JOINT STIFFNESS

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{q}, \mathbf{u})$$

$$\mathbf{K} = \mathbf{K}(\mathbf{q}, \mathbf{u})$$

MACCEPA:
Van Ham et.al, 2007

DLR Hand Arm System:
Grebenstein et.al., 2011

Torque/Stiffness Opt.

- Model of the system dynamics:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \mathbf{u} \in \Omega$$

- Control objective:

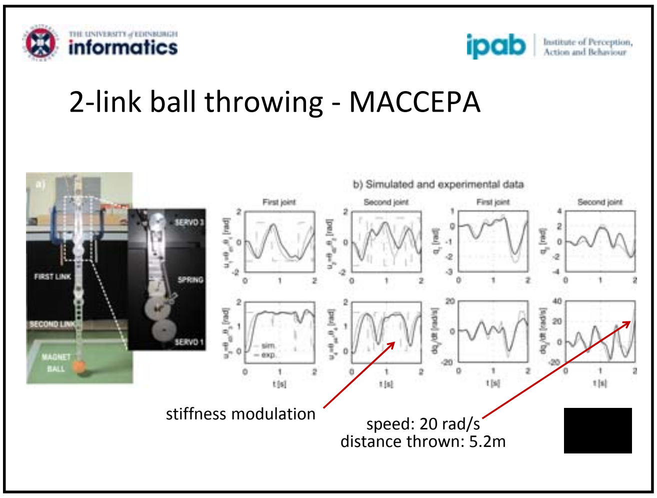
$$J = -d + w\frac{1}{2}\int_0^T \|\mathbf{F}\|^2 dt \rightarrow \min.$$

- Optimal control solution:

$$\mathbf{u}(t, \mathbf{x}) = \mathbf{u}^*(t) + \mathbf{L}^*(t)(\mathbf{x} - \mathbf{x}^*(t))$$

iLQG: Li & Todorov 2007
DDP: Jacobson & Mayne 1970

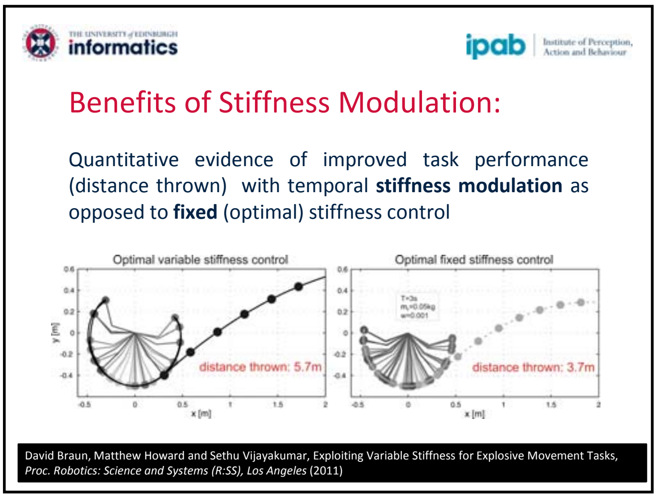
David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)



2-link ball throwing - MACCEPA

stiffness modulation

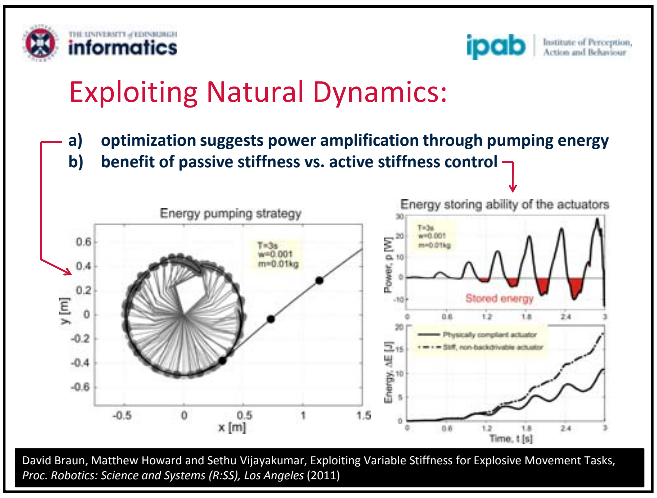
speed: 20 rad/s
distance thrown: 5.2m



Benefits of Stiffness Modulation:

Quantitative evidence of improved task performance (distance thrown) with temporal **stiffness modulation** as opposed to **fixed** (optimal) stiffness control

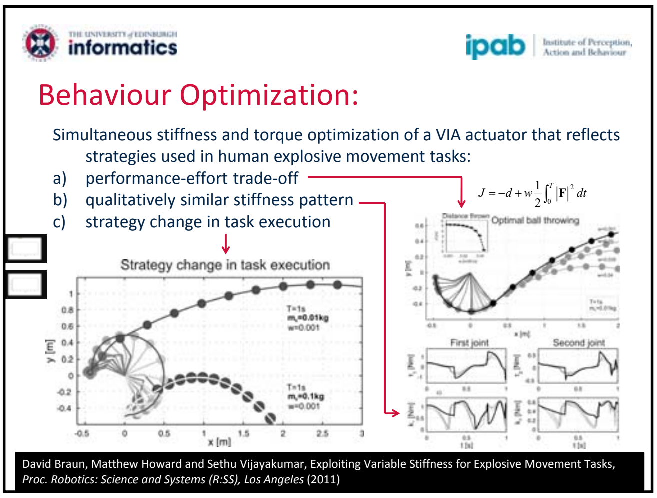
David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)



Exploiting Natural Dynamics:

a) optimization suggests power amplification through pumping energy
b) benefit of passive stiffness vs. active stiffness control

David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)



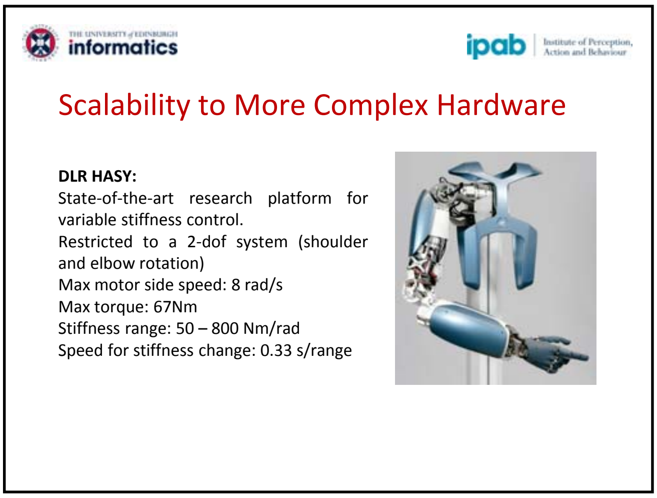
Behaviour Optimization:

Simultaneous stiffness and torque optimization of a VIA actuator that reflects strategies used in human explosive movement tasks:

a) performance-effort trade-off
b) qualitatively similar stiffness pattern
c) strategy change in task execution

$$J = -d + w\frac{1}{2}\int_0^T \|\mathbf{F}\|^2 dt$$

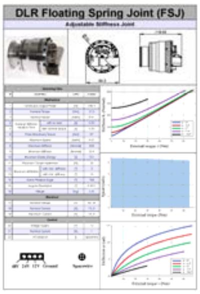
David Braun, Matthew Howard and Sethu Vijayakumar, Exploiting Variable Stiffness for Explosive Movement Tasks, *Proc. Robotics: Science and Systems (R:SS), Los Angeles* (2011)



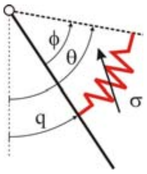
Scalability to More Complex Hardware

DLR HASY:
State-of-the-art research platform for variable stiffness control.
Restricted to a 2-dof system (shoulder and elbow rotation)
Max motor side speed: 8 rad/s
Max torque: 67Nm
Stiffness range: 50 – 800 Nm/rad
Speed for stiffness change: 0.33 s/range

DLR - FSJ



Schematic representation
of the DLR-FSJ



Motor-side positions:

$$\mathbf{q}_2 = [\boldsymbol{\theta},\boldsymbol{\sigma}]^T \in \Re^4$$

Constraint:

$$\phi_{\min}(\sigma) \le \phi \le \phi_{\max}(\sigma)$$

Dealing with Complex Constraints

$$\mathbf{M}_{11}(\mathbf{q}_1)\ddot{\mathbf{q}}_1 + \mathbf{C}_{11}(\mathbf{q}_1,\dot{\mathbf{q}}_1)\dot{\mathbf{q}}_1 + \mathbf{G}_1(\mathbf{q}_1) = \boldsymbol{\tau}_1(\mathbf{q}_1,\mathbf{q}_2)$$

$$\ddot{\mathbf{q}}_2 + 2\boldsymbol{\beta}\dot{\mathbf{q}}_2 + \boldsymbol{\kappa}^2\mathbf{q}_2 = \boldsymbol{\kappa}^2\mathbf{u}$$

Incorporating the constraints:

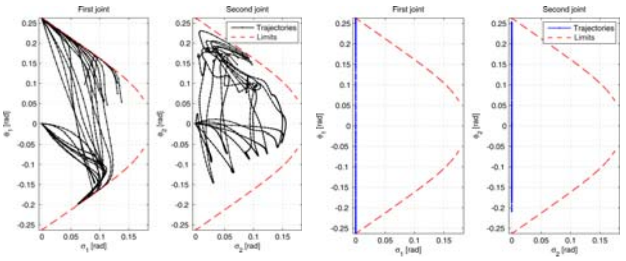
1. Range constraints: $\boldsymbol{\Phi}(\mathbf{q}_1,\mathbf{q}_2) \in \Omega = [\boldsymbol{\Phi}_{\min}(\mathbf{q}_2), \boldsymbol{\Phi}_{\max}(\mathbf{q}_2)]$

   $\mathbf{u} \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}] \Rightarrow \boldsymbol{\Phi}(\mathbf{q}_1,\mathbf{q}_2) \in \Omega$

2. Rate/effort limitations: $\boldsymbol{\kappa} \in [\mathbf{0}, \boldsymbol{\kappa}_{\max}]$

DLR – FSJ: optimisation with state constraints

variable stiffness | fixed stiffness



Spring Length vs Stiffness Modulation

Ball throwing with DLR HASy

David Braun, Florian Petit, Felix Huber, Sami Haddadin, Patrick van der Smagt, Alin Albu-Schaeffer and Sethu Vijayakumar, **Optimal Torque and Stiffness Control in Compliantly Actuated Robots**, *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2012)*, Portugal (2012).

Optimal Variable Impedance

Assume knowledge of **actuator dynamics**
Assume knowledge of **cost** being optimized

- Explosive Movement Tasks (e.g., throwing)
- Periodic Movement Tasks and Temporal Optimization (e.g. walking, brachiation)
- Learning dynamics (OFC-LD)

Jun Nakanishi, Konrad Rawlik and Sethu Vijayakumar, Stiffness and Temporal Optimization in Periodic Movements: An Optimal Control Approach , *Proc. IEEE Intl Conf on Intelligent Robots and Systems (IROS '11) , San Francisco* (2011).

Periodic Movement Control: Issues

Representation

- what is a suitable representation of periodic movement (trajectories, goal)?

Choice of cost function

- how to design a cost function for periodic movement?

Exploitation of natural dynamics

- how to exploit resonance for energy efficient control?
  - optimize frequency (temporal aspect)
  - stiffness tuning

Periodic Movement Representation

WP4

Dynamical system with Fourier basis functions

$$y(t) = r\ \boldsymbol{\psi}^T(\phi)\boldsymbol{\theta} + y_{offset}$$

$$\dot{\phi} = \omega$$

parameters

Fourier basis functions

Fourier basis functions: $\boldsymbol{\psi}(\phi) = [1, \cos\phi, \cdots, \sin(N\phi)]^T$

Fourier coefficients: $\boldsymbol{\theta} = [a_0, a_1, \cdots, b_N]^T$

- scaling of frequency, amplitude and offset is possible
- efficient approximation method to compute Fourier coefficients [Kuhl and Giardina 1982]
- orthogonality properties of basis functions
- cf. Fourier series expansion

$$y(t) = a_0 + \sum_{n=1}^{N}\left(a_n \cos\frac{2n\pi}{T}t + b_n \sin\frac{2n\pi}{T}t\right)$$

Cost Function for Periodic Movements

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Optimization criterion

$$J = \Phi(\mathbf{x}_0, \mathbf{x}_T) + \int_0^T r(\mathbf{x}, \mathbf{u}, t)dt$$

Terminal cost

- ensures periodicity of the trajectory

$$\Phi(\mathbf{x}_0, \mathbf{x}_T) = (\mathbf{x}_T - \mathbf{x}_0)^T\mathbf{Q}_T(\mathbf{x}_T - \mathbf{x}_0)$$

Running cost

- tracking performance and control cost

$$r(\mathbf{x}, \mathbf{u}, t) = (\mathbf{x} - \mathbf{x}_{ref})^T\mathbf{Q}(\mathbf{x} - \mathbf{x}_{ref}) + \mathbf{u}^T\mathbf{R}\mathbf{u}$$

$$\mathbf{x} = [\ y,\ \dot{y}\ ]^T$$

$$y_{ref}(t) = a_0 + \textstyle\sum_{n=1}^{N}(a_n \cos n\omega t + b_n \sin n\omega t)$$

Jun Nakanishi, Konrad Rawlik and Sethu Vijayakumar, Stiffness and Temporal Optimization in Periodic Movements: An Optimal Control Approach , *Proc. IEEE Intl Conf on Intelligent Robots and Systems (IROS '11) , San Francisco* (2011).

Another View of Cost Function

- Running cost: tracking performance and control cost

$$r(\mathbf{x}, \mathbf{u}, t) = (\mathbf{x} - \mathbf{x}_{ref})^T\mathbf{Q}(\mathbf{x} - \mathbf{x}_{ref}) + \mathbf{u}^T\mathbf{R}\mathbf{u}$$

- Augmented plant dynamics with Fourier series based DMPs

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) & (1) \\ y = r\ \boldsymbol{\psi}^T(\phi)\boldsymbol{\theta} + y_{offset} & (2) \\ \dot{\phi} = \omega & (3) \\ \mathbf{z} = \mathbf{x} - \mathbf{y},\ \text{where } \mathbf{y} = [y,\ \dot{y}] & (4) \end{cases}$$

- Reformulated running cost

$$r(\mathbf{z}, \mathbf{u}) = \mathbf{z}^T\mathbf{Q}\mathbf{z} + \mathbf{u}^T\mathbf{R}\mathbf{u}$$

- Find control $\mathbf{u}$ and parameter ω such that plant dynamics (1) should behave like (2) and (3) while min. control cost

Jun Nakanishi, Konrad Rawlik and Sethu Vijayakumar, Stiffness and Temporal Optimization in Periodic Movements: An Optimal Control Approach , *Proc. IEEE Intl Conf on Intelligent Robots and Systems (IROS '11) , San Francisco* (2011).

Temporal Optimization

How do we find the right **temporal duration** in which to optimize a movement ?

Solutions:

- Fix temporal parameters
  ... not optimal
- Time stationary cost
  ... cannot deal with sequential tasks, e.g. via points
- Chain *'first exit time'* controllers
  ... Linear duration cost, not optimal
- Canonical Time Formulation

Canonical Time Formulation

Dynamics: $d\mathbf{x} = f(\mathbf{x}, \mathbf{u})\beta dt + g(\mathbf{x}, \mathbf{u})d\eta$

Cost: $J = \sum_{i=1}^{N}\Phi_i(\mathbf{x}(t_i)) + \int_0^{t_N}\left[r(\mathbf{x}(t)) + \mathbf{u}(t)^T\mathbf{H}\mathbf{u}(t)\right]dt$

n.b. t_i represent *real* time

Introduce change of time $t' = \int_0^t \frac{1}{\beta(s)}ds$

Canonical Time Formulation

Dynamics: $d\mathbf{x} = f(\mathbf{x}, \mathbf{u})\beta dt' + g(\mathbf{x}, \mathbf{u})d\eta'$

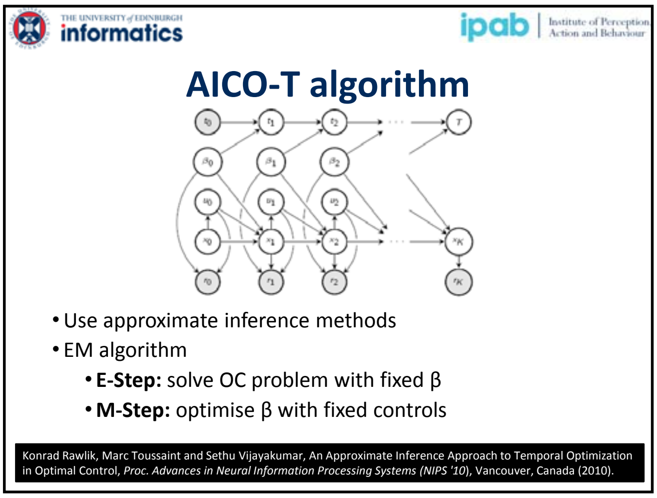
Cost: $J = \sum_{i=1}^{N}\Phi_i\left(\mathbf{x}(\tau^{-1}(t'_i))\right) + \int_0^{\tau^{-1}(t'_N)}\left[r(\mathbf{x}(t)) + \mathbf{u}(t)^T\mathbf{H}\mathbf{u}(t)\right]dt$

$$+ \int_0^{t'_N} c(\beta(s))ds$$

n.b. t'_i now represents *canonical* time

Introduce change of time $t' = \int_0^t \frac{1}{\beta(s)}ds$

Konrad Rawlik, Marc Toussaint and Sethu Vijayakumar, An Approximate Inference Approach to Temporal Optimization in Optimal Control, *Proc. Advances in Neural Information Processing Systems (NIPS '10)*, Vancouver, Canada (2010).

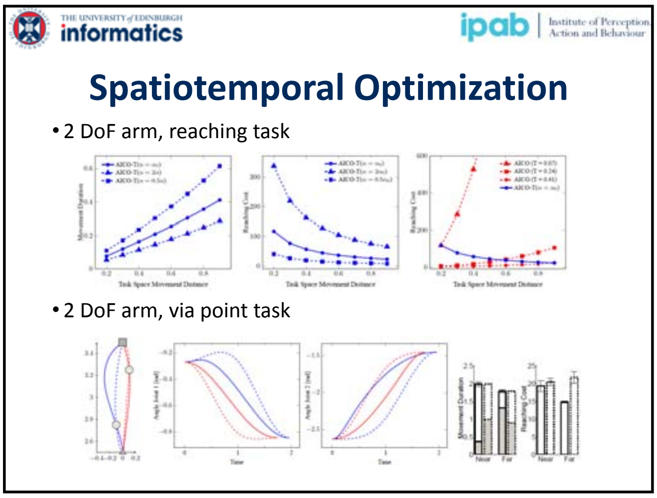


THE UNIVERSITY *of* EDINBURGH informatics | ipab | Institute of Perception Action and Behaviour

AICO-T algorithm

- Use approximate inference methods
- EM algorithm
    - **E-Step:** solve OC problem with fixed β
    - **M-Step:** optimise β with fixed controls

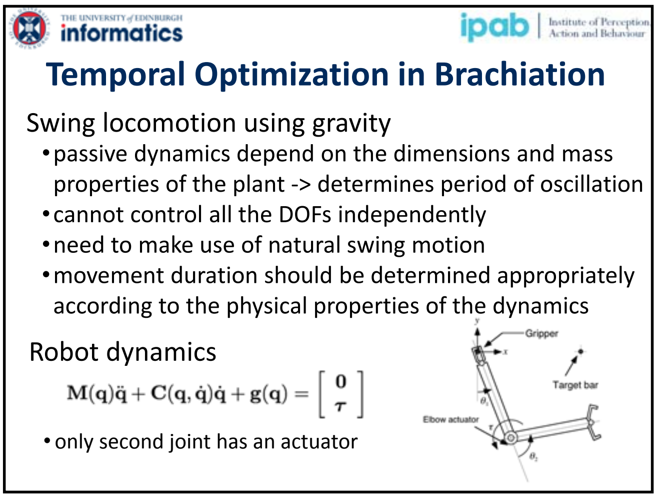
Konrad Rawlik, Marc Toussaint and Sethu Vijayakumar, An Approximate Inference Approach to Temporal Optimization in Optimal Control, *Proc. Advances in Neural Information Processing Systems (NIPS '10)*, Vancouver, Canada (2010).



THE UNIVERSITY *of* EDINBURGH informatics | ipab | Institute of Perception Action and Behaviour

Spatiotemporal Optimization

- 2 DoF arm, reaching task
- 2 DoF arm, via point task



THE UNIVERSITY *of* EDINBURGH informatics | ipab | Institute of Perception Action and Behaviour

Temporal Optimization in Brachiation

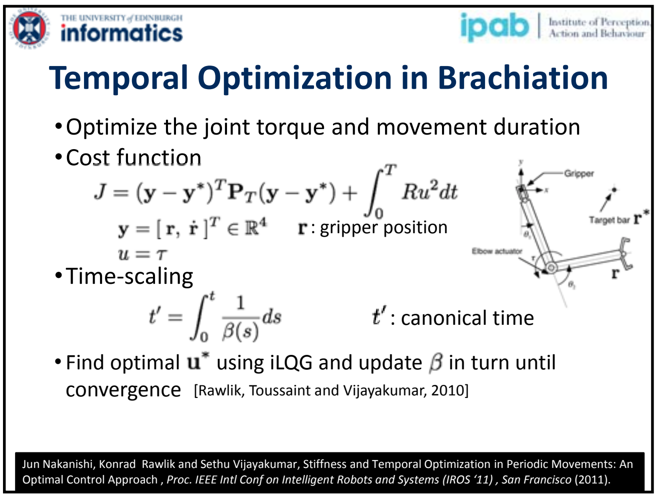
Swing locomotion using gravity

- passive dynamics depend on the dimensions and mass properties of the plant -> determines period of oscillation
- cannot control all the DOFs independently
- need to make use of natural swing motion
- movement duration should be determined appropriately according to the physical properties of the dynamics

Robot dynamics

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix}$$

- only second joint has an actuator



THE UNIVERSITY *of* EDINBURGH informatics | ipab | Institute of Perception Action and Behaviour

Temporal Optimization in Brachiation

- Optimize the joint torque and movement duration
- Cost function

$$J = (\mathbf{y} - \mathbf{y}^*)^T \mathbf{P}_T (\mathbf{y} - \mathbf{y}^*) + \int_0^T R u^2 dt$$

$\mathbf{y} = [\, \mathbf{r},\ \dot{\mathbf{r}} \,]^T \in \mathbb{R}^4$ $\mathbf{r}$: gripper position

$u = \tau$

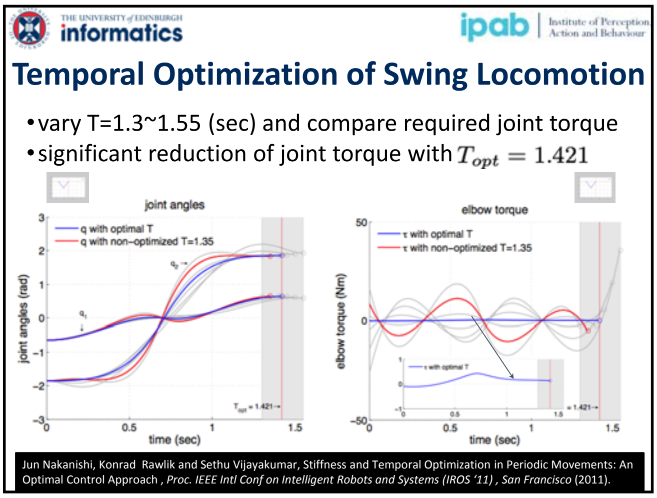
- Time-scaling

$$t' = \int_0^t \frac{1}{\beta(s)} ds$$

t' : canonical time

- Find optimal $\mathbf{u}^*$ using iLQG and update β in turn until convergence [Rawlik, Toussaint and Vijayakumar, 2010]

Jun Nakanishi, Konrad Rawlik and Sethu Vijayakumar, Stiffness and Temporal Optimization in Periodic Movements: An Optimal Control Approach , *Proc. IEEE Intl Conf on Intelligent Robots and Systems (IROS '11) , San Francisco* (2011).

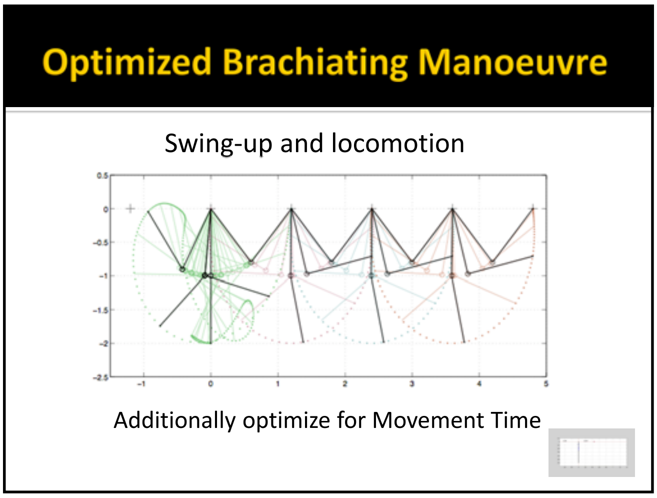


THE UNIVERSITY *of* EDINBURGH informatics | ipab | Institute of Perception Action and Behaviour

Temporal Optimization of Swing Locomotion

- vary T=1.3~1.55 (sec) and compare required joint torque
- significant reduction of joint torque with $T_{opt} = 1.421$

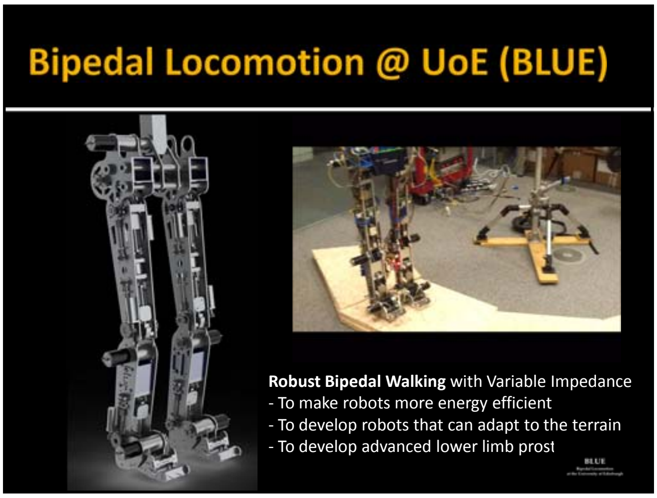
Jun Nakanishi, Konrad Rawlik and Sethu Vijayakumar, Stiffness and Temporal Optimization in Periodic Movements: An Optimal Control Approach , *Proc. IEEE Intl Conf on Intelligent Robots and Systems (IROS '11) , San Francisco* (2011).



Optimized Brachiating Manoeuvre

Swing-up and locomotion

Additionally optimize for Movement Time



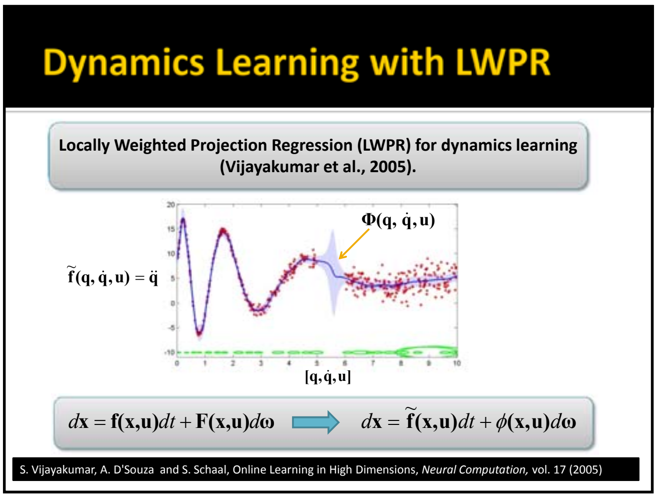
Bipedal Locomotion @ UoE (BLUE)

Robust Bipedal Walking with Variable Impedance
- To make robots more energy efficient
- To develop robots that can adapt to the terrain
- To develop advanced lower limb prost

Variable Impedance Policies
-- through Stochastic Optimization

~~Assume knowledge of actuator dynamics~~
Assume knowledge of **cost** being optimized

- Explosive Movement Tasks (e.g., throwing)
- Periodic Movement Tasks and Temporal Optimization (e.g. walking, brachiation)
- Learning dynamics (OFC-LD)



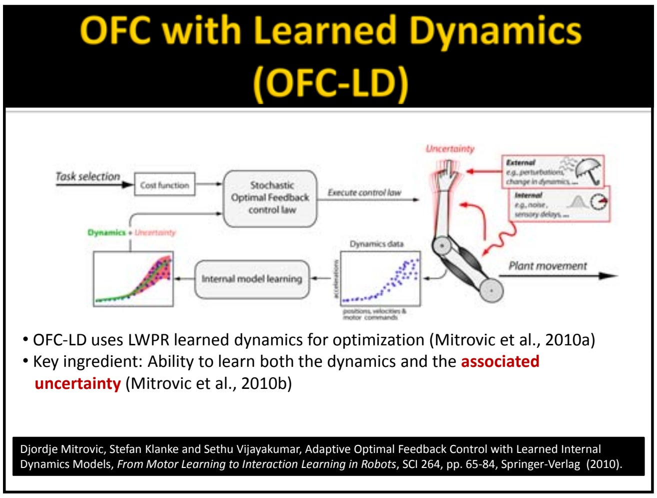
Dynamics Learning with LWPR

Locally Weighted Projection Regression (LWPR) for dynamics learning (Vijayakumar et al., 2005).

$\tilde{\mathbf{f}}(\mathbf{q},\dot{\mathbf{q}},\mathbf{u}) = \ddot{\mathbf{q}}$

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},\mathbf{u})dt + \mathbf{F}(\mathbf{x},\mathbf{u})d\omega \Longrightarrow d\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x},\mathbf{u})dt + \phi(\mathbf{x},\mathbf{u})d\omega$$

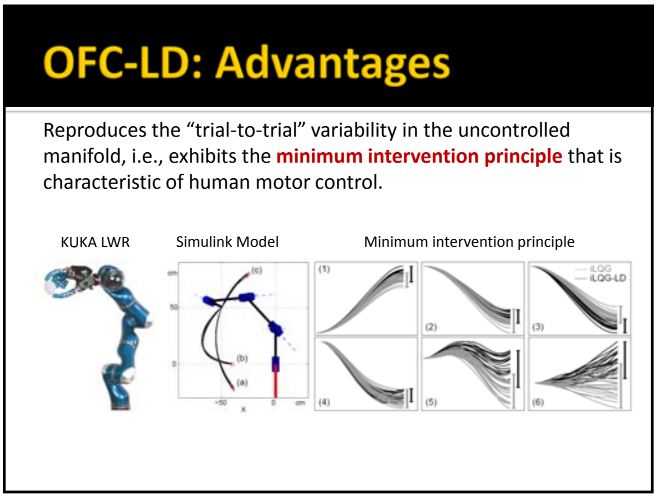
S. Vijayakumar, A. D'Souza and S. Schaal, Online Learning in High Dimensions, *Neural Computation*, vol. 17 (2005)



OFC with Learned Dynamics (OFC-LD)

- OFC-LD uses LWPR learned dynamics for optimization (Mitrovic et al., 2010a)
- Key ingredient: Ability to learn both the dynamics and the **associated uncertainty** (Mitrovic et al., 2010b)

Djordje Mitrovic, Stefan Klanke and Sethu Vijayakumar, Adaptive Optimal Feedback Control with Learned Internal Dynamics Models, *From Motor Learning to Interaction Learning in Robots*, SCI 264, pp. 65-84, Springer-Verlag (2010).



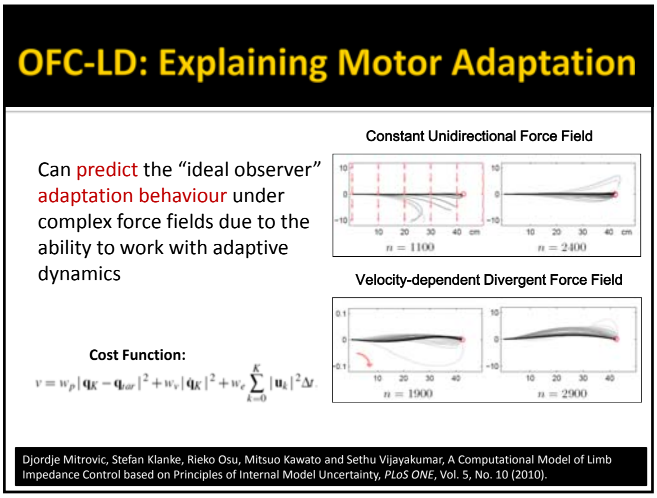
OFC-LD: Advantages

Reproduces the "trial-to-trial" variability in the uncontrolled manifold, i.e., exhibits the **minimum intervention principle** that is characteristic of human motor control.

KUKA LWR

Simulink Model

Minimum intervention principle



OFC-LD: Explaining Motor Adaptation

Can predict the "ideal observer" adaptation behaviour under complex force fields due to the ability to work with adaptive dynamics

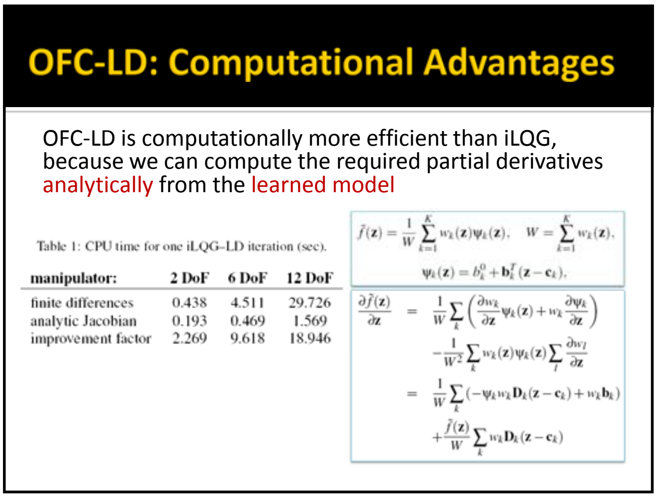
Cost Function:

$$v = w_p|\mathbf{q}_K - \mathbf{q}_{tar}|^2 + w_v|\dot{\mathbf{q}}_K|^2 + w_e\sum_{k=0}^{K}|\mathbf{u}_k|^2\Delta t.$$

Constant Unidirectional Force Field

Velocity-dependent Divergent Force Field

Djordje Mitrovic, Stefan Klanke, Rieko Osu, Mitsuo Kawato and Sethu Vijayakumar, A Computational Model of Limb Impedance Control based on Principles of Internal Model Uncertainty, *PLoS ONE*, Vol. 5, No. 10 (2010).



OFC-LD: Computational Advantages

OFC-LD is computationally more efficient than iLQG, because we can compute the required partial derivatives analytically from the learned model

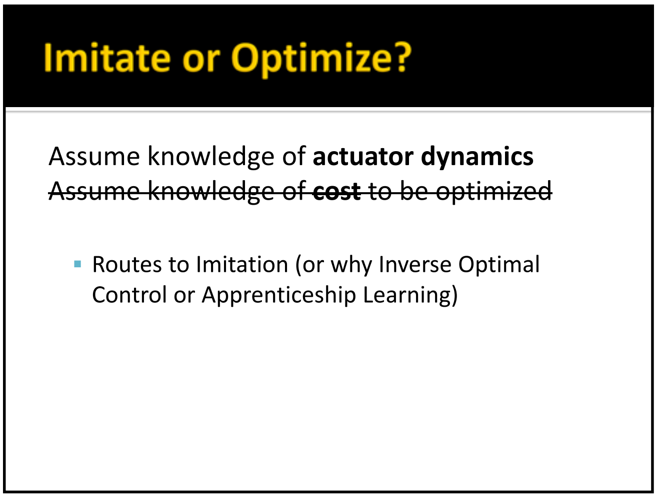
Table 1: CPU time for one iLQG–LD iteration (sec).

manipulator:	2 DoF	6 DoF	12 DoF
finite differences	0.438	4.511	29.726
analytic Jacobian	0.193	0.469	1.569
improvement factor	2.269	9.618	18.946

$$\tilde{f}(\mathbf{z}) = \frac{1}{W}\sum_{k=1}^{K} w_k(\mathbf{z})\psi_k(\mathbf{z}), \quad W = \sum_{k=1}^{K} w_k(\mathbf{z}),$$

$$\psi_k(\mathbf{z}) = b_k^0 + \mathbf{b}_k^T(\mathbf{z}-\mathbf{c}_k),$$

$$\begin{aligned}\frac{\partial \tilde{f}(\mathbf{z})}{\partial \mathbf{z}} &= \frac{1}{W}\sum_k \left(\frac{\partial w_k}{\partial \mathbf{z}}\psi_k(\mathbf{z}) + w_k \frac{\partial \psi_k}{\partial \mathbf{z}}\right) \\ &\quad -\frac{1}{W^2}\sum_k w_k(\mathbf{z})\psi_k(\mathbf{z})\sum_l \frac{\partial w_l}{\partial \mathbf{z}} \\ &= \frac{1}{W}\sum_k (-\psi_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k) + w_k \mathbf{b}_k) \\ &\quad +\frac{\tilde{f}(\mathbf{z})}{W}\sum_k w_k \mathbf{D}_k(\mathbf{z}-\mathbf{c}_k)\end{aligned}$$

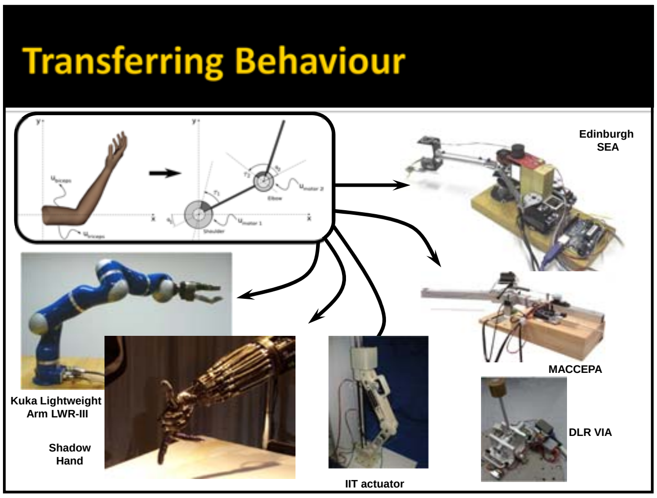


Imitate or Optimize?

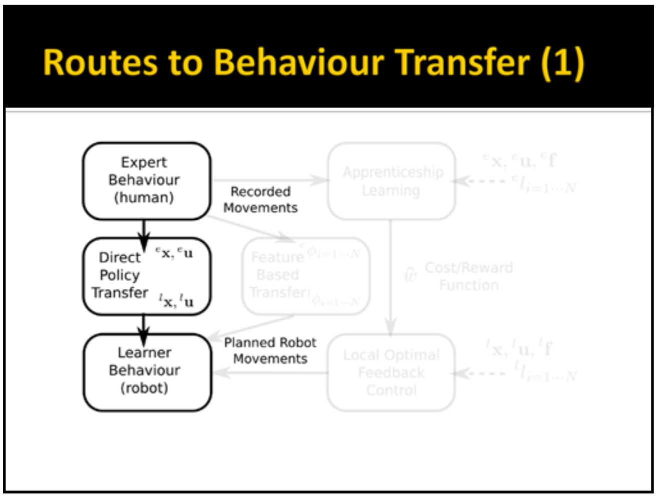
Assume knowledge of **actuator dynamics**

~~Assume knowledge of **cost** to be optimized~~

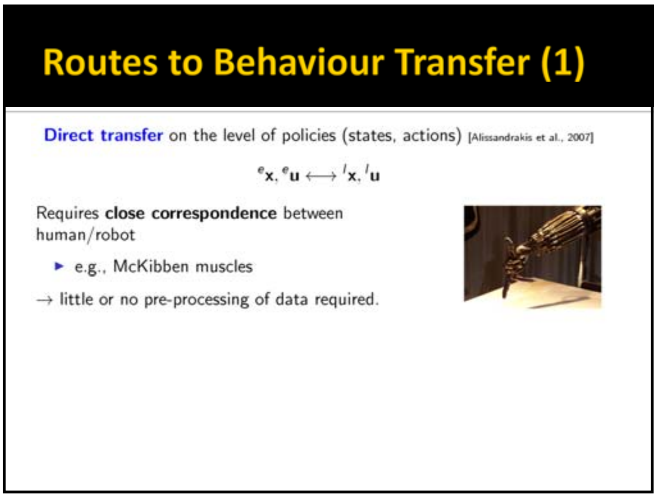
- Routes to Imitation (or why Inverse Optimal Control or Apprenticeship Learning)



Transferring Behaviour



Routes to Behaviour Transfer (1)



Routes to Behaviour Transfer (1)

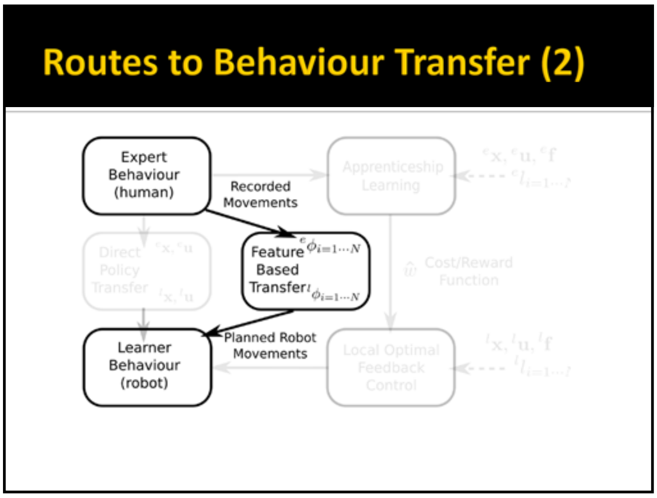
Direct transfer on the level of policies (states, actions) [Alissandrakis et al., 2007]

$$^e\mathbf{x}, {}^e\mathbf{u} \longleftrightarrow {}^l\mathbf{x}, {}^l\mathbf{u}$$

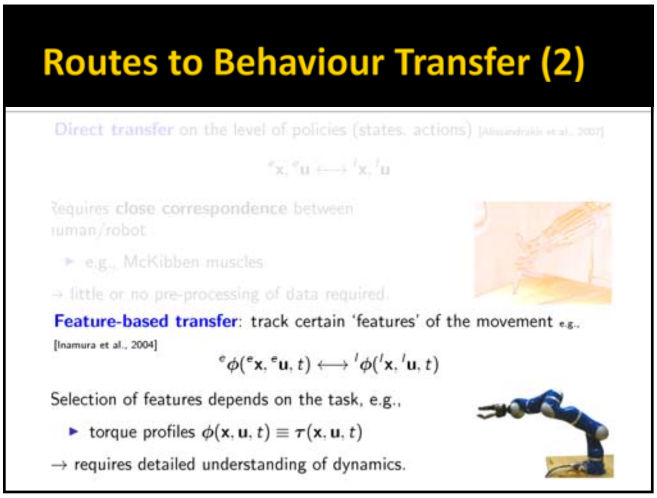
Requires **close correspondence** between human/robot

- e.g., McKibben muscles

→ little or no pre-processing of data required.



Routes to Behaviour Transfer (2)



Routes to Behaviour Transfer (2)

Direct transfer on the level of policies (states, actions) [Alissandrakis et al., 2007]

$$^{e}x, {}^{e}u \longleftrightarrow {}^{l}x, {}^{l}u$$

Requires **close correspondence** between human/robot

- e.g., McKibben muscles

→ little or no pre-processing of data required.

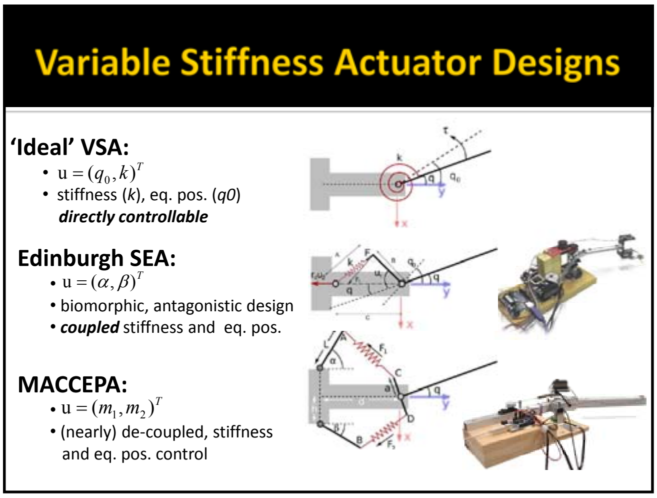
Feature-based transfer: track certain 'features' of the movement e.g., [Inamura et al., 2004]

$$^{e}\phi(^{e}\mathbf{x}, {}^{e}\mathbf{u}, t) \longleftrightarrow {}^{l}\phi(^{l}\mathbf{x}, {}^{l}\mathbf{u}, t)$$

Selection of features depends on the task, e.g.,

- torque profiles $\phi(\mathbf{x}, \mathbf{u}, t) \equiv \tau(\mathbf{x}, \mathbf{u}, t)$

→ requires detailed understanding of dynamics.

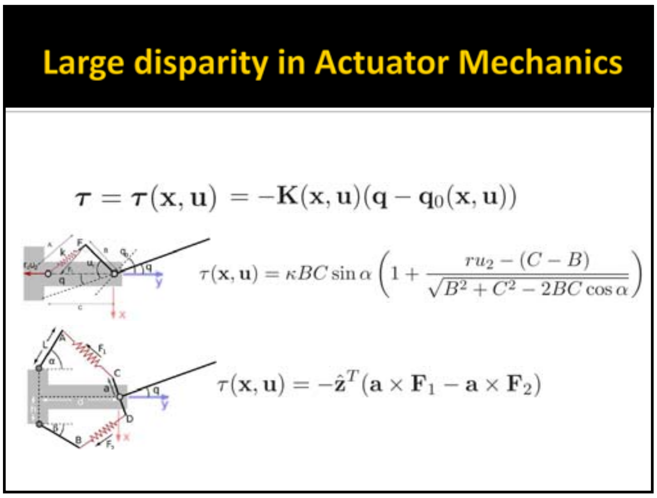


Variable Stiffness Actuator Designs

'Ideal' VSA:
- $u = (q_0, k)^T$
- stiffness (*k*), eq. pos. (*q0*) ***directly controllable***

Edinburgh SEA:
- $u = (\alpha, \beta)^T$
- biomorphic, antagonistic design
- ***coupled*** stiffness and eq. pos.

MACCEPA:
- $u = (m_1, m_2)^T$
- (nearly) de-coupled, stiffness and eq. pos. control

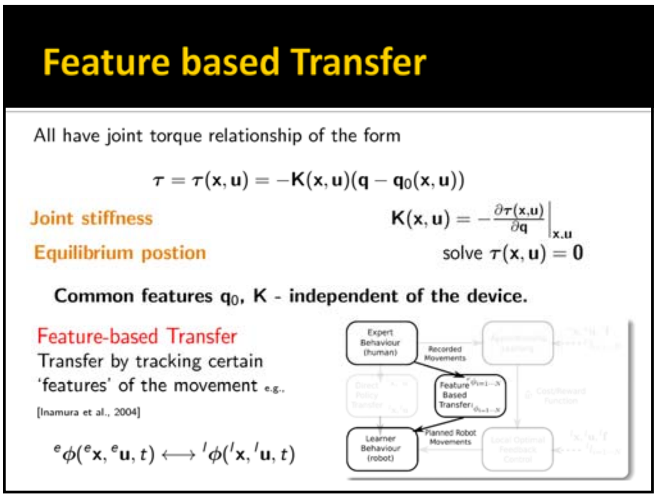


Large disparity in Actuator Mechanics

$$\tau = \tau(\mathbf{x}, \mathbf{u}) = -\mathbf{K}(\mathbf{x}, \mathbf{u})(\mathbf{q} - \mathbf{q}_0(\mathbf{x}, \mathbf{u}))$$

$$\tau(\mathbf{x}, \mathbf{u}) = \kappa BC \sin\alpha \left(1 + \frac{ru_2 - (C-B)}{\sqrt{B^2 + C^2 - 2BC\cos\alpha}}\right)$$

$$\tau(\mathbf{x}, \mathbf{u}) = -\hat{\mathbf{z}}^T(\mathbf{a} \times \mathbf{F}_1 - \mathbf{a} \times \mathbf{F}_2)$$



Feature based Transfer

All have joint torque relationship of the form

$$\tau = \tau(\mathbf{x}, \mathbf{u}) = -\mathbf{K}(\mathbf{x}, \mathbf{u})(\mathbf{q} - \mathbf{q}_0(\mathbf{x}, \mathbf{u}))$$

Joint stiffness $\mathbf{K}(\mathbf{x}, \mathbf{u}) = -\frac{\partial \tau(\mathbf{x},\mathbf{u})}{\partial \mathbf{q}}\Big|_{\mathbf{x},\mathbf{u}}$

Equilibrium postion solve $\tau(\mathbf{x}, \mathbf{u}) = \mathbf{0}$

Common features $\mathbf{q}_0$, K - independent of the device.

Feature-based Transfer

Transfer by tracking certain 'features' of the movement e.g.,

[Inamura et al., 2004]

$$^{e}\phi(^{e}\mathbf{x}, {}^{e}\mathbf{u}, t) \longleftrightarrow {}^{l}\phi(^{l}\mathbf{x}, {}^{l}\mathbf{u}, t)$$

Feature based Transfer

Given

$$\mathbf{q}_0 = \mathbf{q}_0(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^n \quad \text{and} \quad \mathbf{k} = \mathbf{k}(\mathbf{x}, \mathbf{u}) = vec(\mathbf{K}) \in \mathbb{R}^{n^2}$$

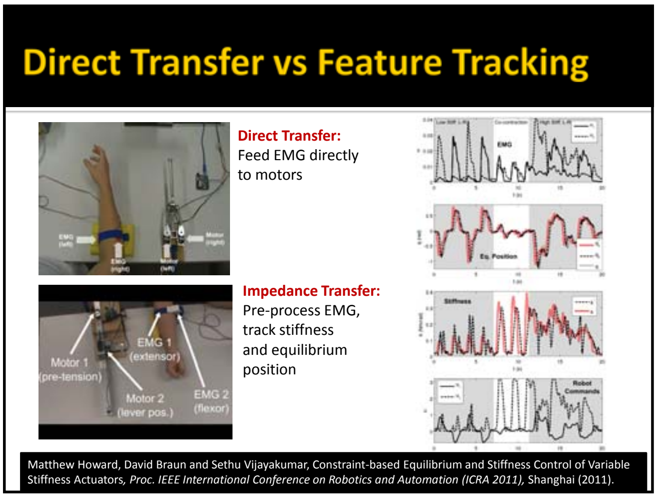
Take derivatives

$$\dot{\mathbf{q}}_0 = \mathbf{J}_{\mathbf{q}_0}(\mathbf{x}, \mathbf{u})\dot{\mathbf{u}} + \mathbf{P}_{\mathbf{q}_0}(\mathbf{x}, \mathbf{u})\dot{\mathbf{x}}, \qquad \dot{\mathbf{k}} = \mathbf{J}_{\mathbf{k}}\ (\mathbf{x}, \mathbf{u})\dot{\mathbf{u}} + \mathbf{P}_{\mathbf{k}}\ (\mathbf{x}, \mathbf{u})\dot{\mathbf{x}},$$

Constrain changes in u according to

$$\dot{\mathbf{u}} = \mathbf{J}(\mathbf{x}, \mathbf{u})^{\dagger}\dot{\mathbf{r}} + (\mathbf{I} - \mathbf{J}(\mathbf{x}, \mathbf{u})^{\dagger}\mathbf{J}(\mathbf{x}, \mathbf{u}))\mathbf{a}$$

- **r** is our task space ($\mathbf{q}_0$, **k**, or both)
- **J** is the appropriate Jacobian ($\mathbf{J}_{\mathbf{q}_0}$, $\mathbf{J}_{\mathbf{k}}$, or both)
- **a** is an arbitrary redundancy term.



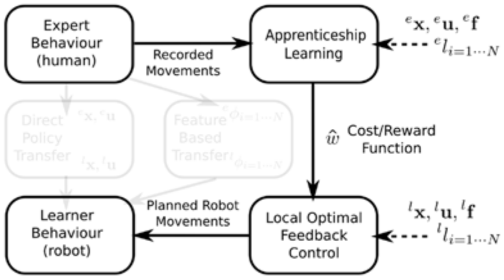
Direct Transfer vs Feature Tracking

Direct Transfer:
Feed EMG directly to motors

Impedance Transfer:
Pre-process EMG, track stiffness and equilibrium position

Matthew Howard, David Braun and Sethu Vijayakumar, Constraint-based Equilibrium and Stiffness Control of Variable Stiffness Actuators, *Proc. IEEE International Conference on Robotics and Automation (ICRA 2011),* Shanghai (2011).

Routes to Behaviour Transfer (3)



Cost Functions for Movement Plans

Multiplicative Weights Apprenticeship Learning [Syed et al., 2008]

Inverse optimal control method

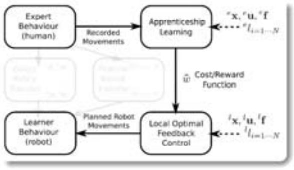
- We are given $^e\mathbf{f}$, ^eX, ^eU . . . **we seek** $^eJ(\mathbf{x}, \mathbf{u}, t)$

Key Assumption

$$^eJ = \sum_{i=1}^{n_T} w_i \, {^eh_i}({^e\mathbf{x}}(T))$$

$$+\sum_{i=n_T}^{N} w_i \int_0^T {^el_i}({^e\mathbf{x}}, {^e\mathbf{u}}, t)\, dt$$

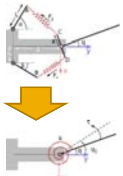
with $^eh_i(\cdot), {^el_i}(\cdot)$ **known.**

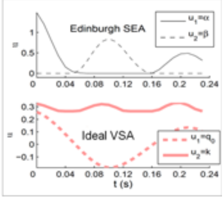


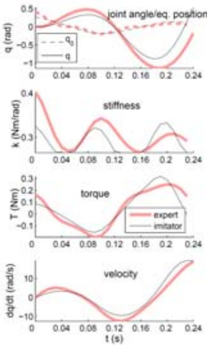
Iterative Approach

- Solve **forward optimisation** under current estimate of **w**
- Update $\hat{\mathbf{w}}$ by comparing **value functions**

Transferring Behaviour: Different Actuators





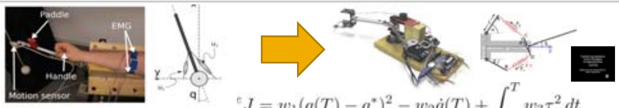


Transfer ball hitting task across different VIAs:

Very different command sequences due to different actuation

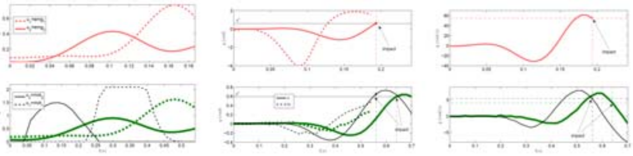
Optimal impedance control strategy very similar across plants

Imitating Human Hitting



$$^eJ = w_1(q(T) - q^*)^2 - w_2\dot{q}(T) + \int_0^T w_3\tau^2\, dt$$

- **Direct imitation:** lower velocity at time of impact, less powerful hit
- **Apprenticeship learning:** movement is optimised to robot dynamics, ball is hit further

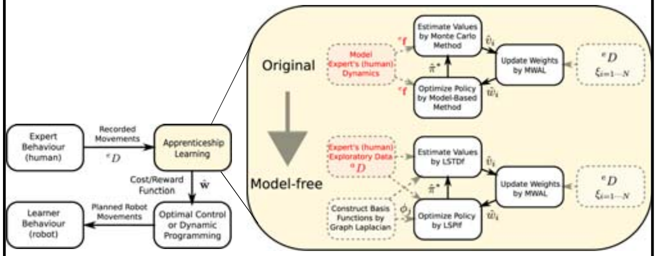


M. Howard, D. Mitrovic & S. Vijayakumar, Transferring Impedance Control Strategies Between Heterogeneous Systems via Apprenticeship Learning, *Proc. IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA (2010)).

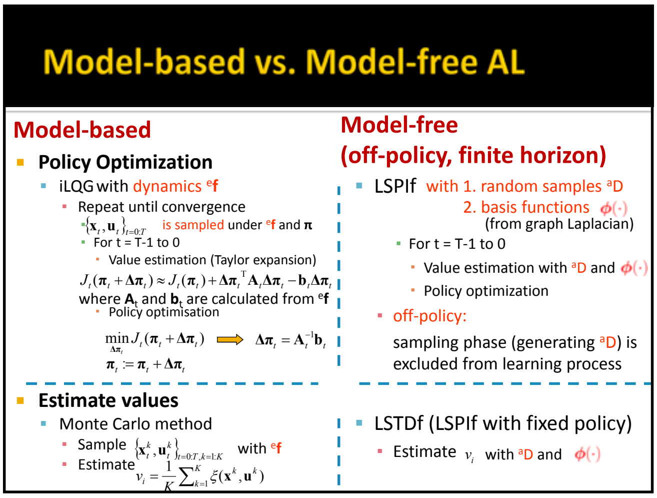
Need for Model Free Methods

- Model-based transfer of human behavior has relied on demonstrator's dynamics: in most practical settings, such models fail to capture
  - the complex, non-linear dynamics of the human musculoskeletal system
  - inconsistencies between modeling assumptions and the configuration and placement of measurement apparatus

Model-free Transfer



▪	Original	Monte Carlo method and model-based method on MWAL Requires: (human) dynamics model $^e\mathbf{f}$
▪	Model-free	LSTDf and LSPIf combined on MWAL Requires: exploratory data ^aD instead of using dynamics model



Model-based vs. Model-free AL

Model-based

- **Policy Optimization**
  - iLQG with dynamics ${}^{e}\mathbf{f}$
    - Repeat until convergence
      - $\{\mathbf{x}_t, \mathbf{u}_t\}_{t=0:T}$ is sampled under ${}^{e}\mathbf{f}$ and $\boldsymbol{\pi}$
      - For t = T-1 to 0
        - Value estimation (Taylor expansion)

$$J_t(\boldsymbol{\pi}_t + \Delta\boldsymbol{\pi}_t) \approx J_t(\boldsymbol{\pi}_t) + \Delta\boldsymbol{\pi}_t^{\mathrm{T}} \mathbf{A}_t \Delta\boldsymbol{\pi}_t - \mathbf{b}_t \Delta\boldsymbol{\pi}_t$$

where $\mathbf{A}_t$ and $\mathbf{b}_t$ are calculated from ${}^{e}\mathbf{f}$

        - Policy optimisation

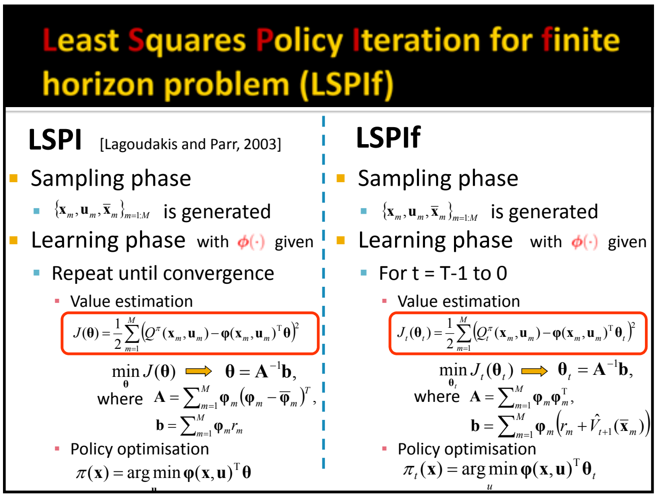
$$\min_{\Delta\boldsymbol{\pi}_t} J_t(\boldsymbol{\pi}_t + \Delta\boldsymbol{\pi}_t) \Rightarrow \Delta\boldsymbol{\pi}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$$

$$\boldsymbol{\pi}_t := \boldsymbol{\pi}_t + \Delta\boldsymbol{\pi}_t$$

- **Estimate values**
  - Monte Carlo method
    - Sample $\{\mathbf{x}_t^k, \mathbf{u}_t^k\}_{t=0:T,k=1:K}$ with ${}^{e}\mathbf{f}$
    - Estimate $v_i = \frac{1}{K}\sum_{k=1}^{K} \xi(\mathbf{x}^k, \mathbf{u}^k)$

Model-free (off-policy, finite horizon)

- LSPIf with 1. random samples ^{a}D
  2. basis functions $\phi(\cdot)$
  (from graph Laplacian)
    - For t = T-1 to 0
      - Value estimation with ^{a}D and $\phi(\cdot)$
      - Policy optimization
  - off-policy:
    sampling phase (generating ^{a}D) is excluded from learning process
- LSTDf (LSPIf with fixed policy)
  - Estimate v_i with ^{a}D and $\phi(\cdot)$



Least Squares Policy Iteration for finite horizon problem (LSPIf)

LSPI [Lagoudakis and Parr, 2003]

- Sampling phase
  - $\{\mathbf{x}_m, \mathbf{u}_m, \bar{\mathbf{x}}_m\}_{m=1:M}$ is generated
- Learning phase with $\phi(\cdot)$ given
  - Repeat until convergence
    - Value estimation

$$\boxed{J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{m=1}^{M}\left(Q^{\pi}(\mathbf{x}_m, \mathbf{u}_m) - \boldsymbol{\varphi}(\mathbf{x}_m, \mathbf{u}_m)^{\mathrm{T}}\boldsymbol{\theta}\right)^2}$$

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \Rightarrow \boldsymbol{\theta} = \mathbf{A}^{-1}\mathbf{b},$$

where $\mathbf{A} = \sum_{m=1}^{M} \boldsymbol{\varphi}_m (\boldsymbol{\varphi}_m - \bar{\boldsymbol{\varphi}}_m)^T,$

$\mathbf{b} = \sum_{m=1}^{M} \boldsymbol{\varphi}_m r_m$

    - Policy optimisation

$$\pi(\mathbf{x}) = \arg\min_{\mathbf{u}} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{u})^{\mathrm{T}}\boldsymbol{\theta}$$

LSPIf

- Sampling phase
  - $\{\mathbf{x}_m, \mathbf{u}_m, \bar{\mathbf{x}}_m\}_{m=1:M}$ is generated
- Learning phase with $\phi(\cdot)$ given
  - For t = T-1 to 0
    - Value estimation

$$\boxed{J_t(\boldsymbol{\theta}_t) = \frac{1}{2}\sum_{m=1}^{M}\left(Q_t^{\pi}(\mathbf{x}_m, \mathbf{u}_m) - \boldsymbol{\varphi}(\mathbf{x}_m, \mathbf{u}_m)^{\mathrm{T}}\boldsymbol{\theta}_t\right)^2}$$

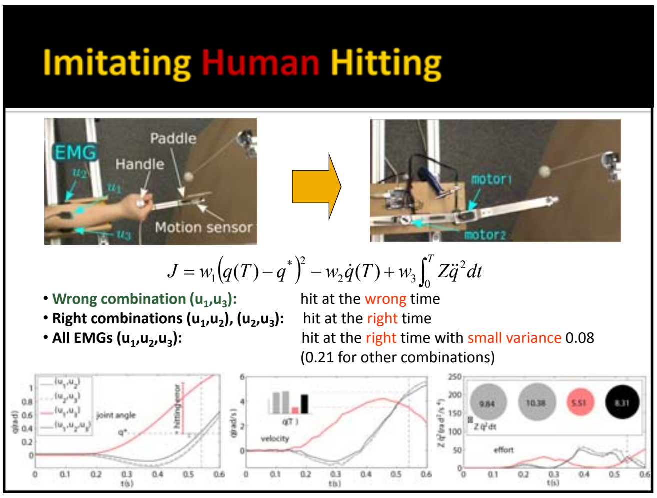
$$\min_{\boldsymbol{\theta}_t} J_t(\boldsymbol{\theta}_t) \Rightarrow \boldsymbol{\theta}_t = \mathbf{A}^{-1}\mathbf{b},$$

where $\mathbf{A} = \sum_{m=1}^{M} \boldsymbol{\varphi}_m \boldsymbol{\varphi}_m^{\mathrm{T}},$

$\mathbf{b} = \sum_{m=1}^{M} \boldsymbol{\varphi}_m \left(r_m + \hat{V}_{t+1}(\bar{\mathbf{x}}_m)\right)$

    - Policy optimisation

$$\pi_t(\mathbf{x}) = \arg\min_{u} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{u})^{\mathrm{T}}\boldsymbol{\theta}_t$$



Imitating Human Hitting

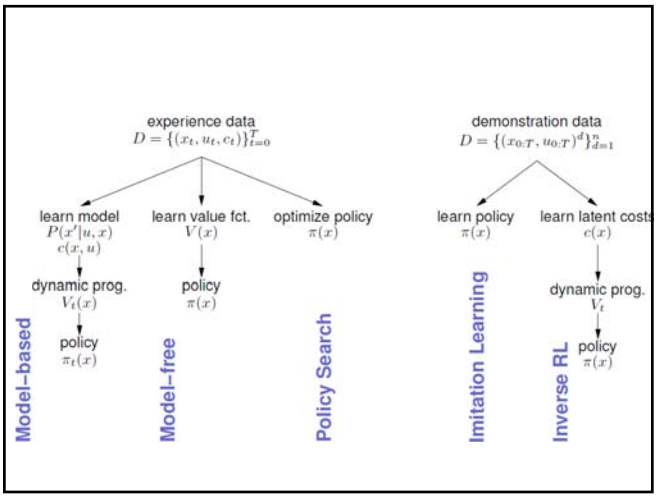
$$J = w_1\left(q(T) - q^*\right)^2 - w_2\dot{q}(T) + w_3\int_0^T Z\ddot{q}^2 dt$$

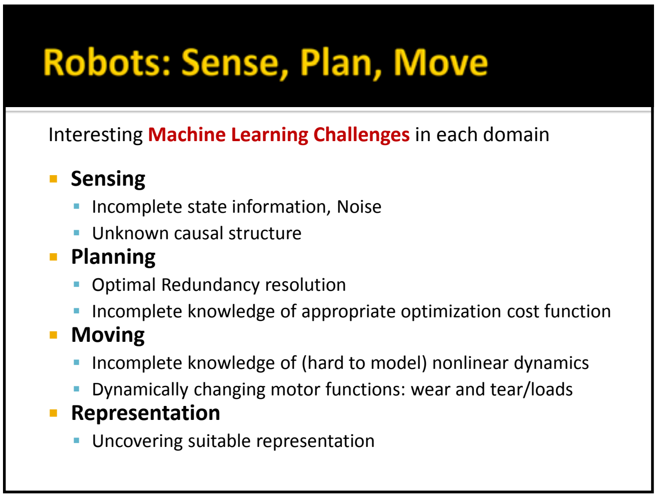
- **Wrong combination (u_1,u_3):** hit at the wrong time
- **Right combinations (u_1,u_2), (u_2,u_3):** hit at the right time
- **All EMGs (u_1,u_2,u_3):** hit at the right time with small variance 0.08 (0.21 for other combinations)

Heirarchical Planning in Topology Spaces

- Generalize
- Scale and Re-plan
- Deal with Dynamic Constraints

Dmitry Zarubin, Vladimir Ivan, Taku Komura, Marc Toussaint and Sethu Vijayakumar, Hierarchical Motion Planning in Topological Spaces, Proc. Robotics: Science and Systems (R:SS 2012), Sydney, Australia (2012).





Robots: Sense, Plan, Move

Interesting **Machine Learning Challenges** in each domain

- **Sensing**
  - Incomplete state information, Noise
  - Unknown causal structure
- **Planning**
  - Optimal Redundancy resolution
  - Incomplete knowledge of appropriate optimization cost function
- **Moving**
  - Incomplete knowledge of (hard to model) nonlinear dynamics
  - Dynamically changing motor functions: wear and tear/loads
- **Representation**
  - Uncovering suitable representation