Neural and probabilistic learning methods for robotics and other domains

Tutorial at SoftNet 2018

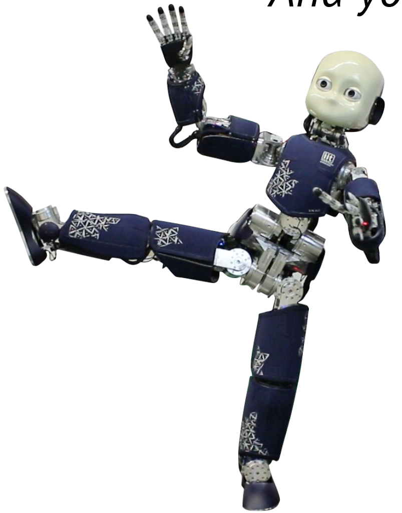
Nice, October 14th, 2018
Prof. Dr. Elmar Rueckert

latest updated July 23rd 2018

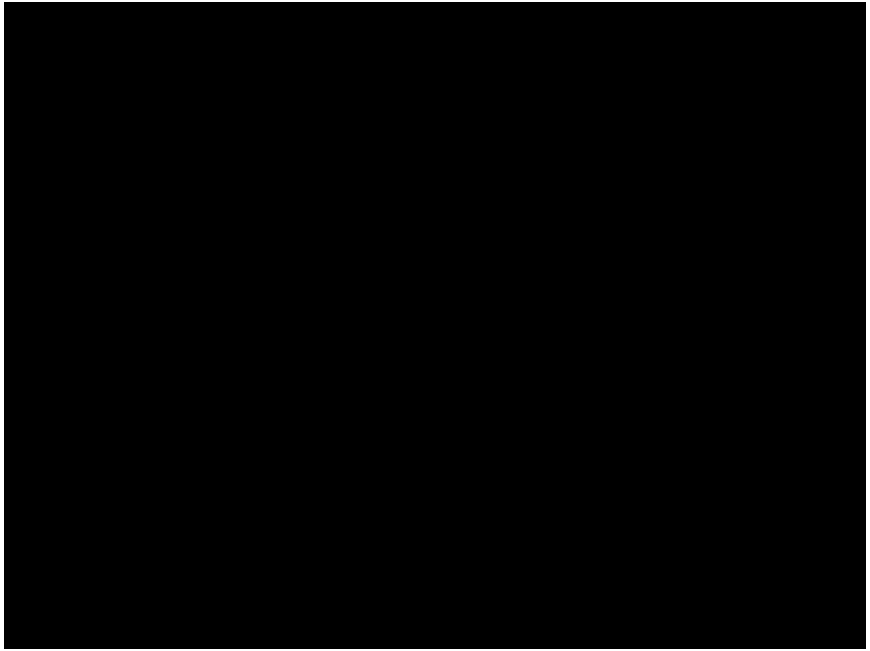
Introduction & Motivation

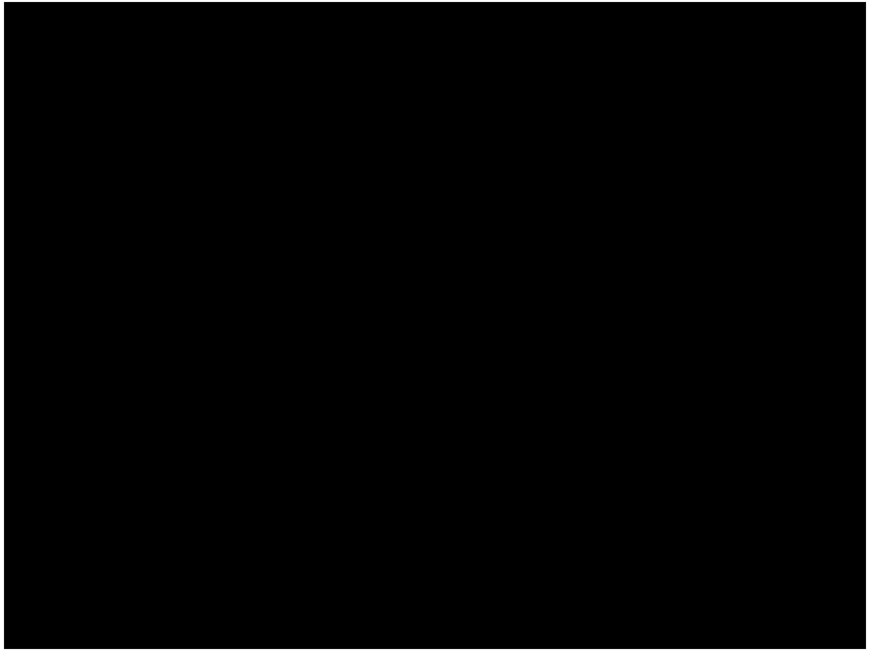
Humanoid robots are among the most complex machines on earth.

And you will learn here how to build, teach and program them.



Challenges in motor skill learning

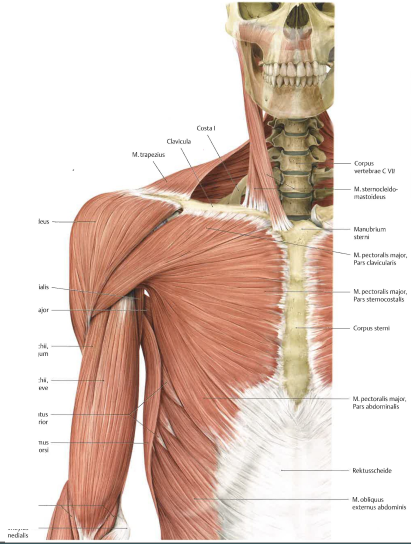


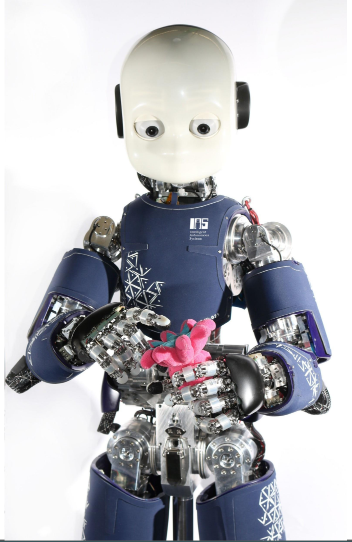


More than robotics …

The challenges in understanding humans and in building intelligent humanoids are converging!

~ 700 muscles
~ 100 joints
~ 100 x 10^6 photo receptors
~ 10^2 FA-I receptors per fingertip





- 53 degrees of freedom
- 4 force/torque sensors
- 1.8 x 10^6 photo receptors
~ 2000 tactile sensors

In **humans** we suffer from noise, accuracy, delays.

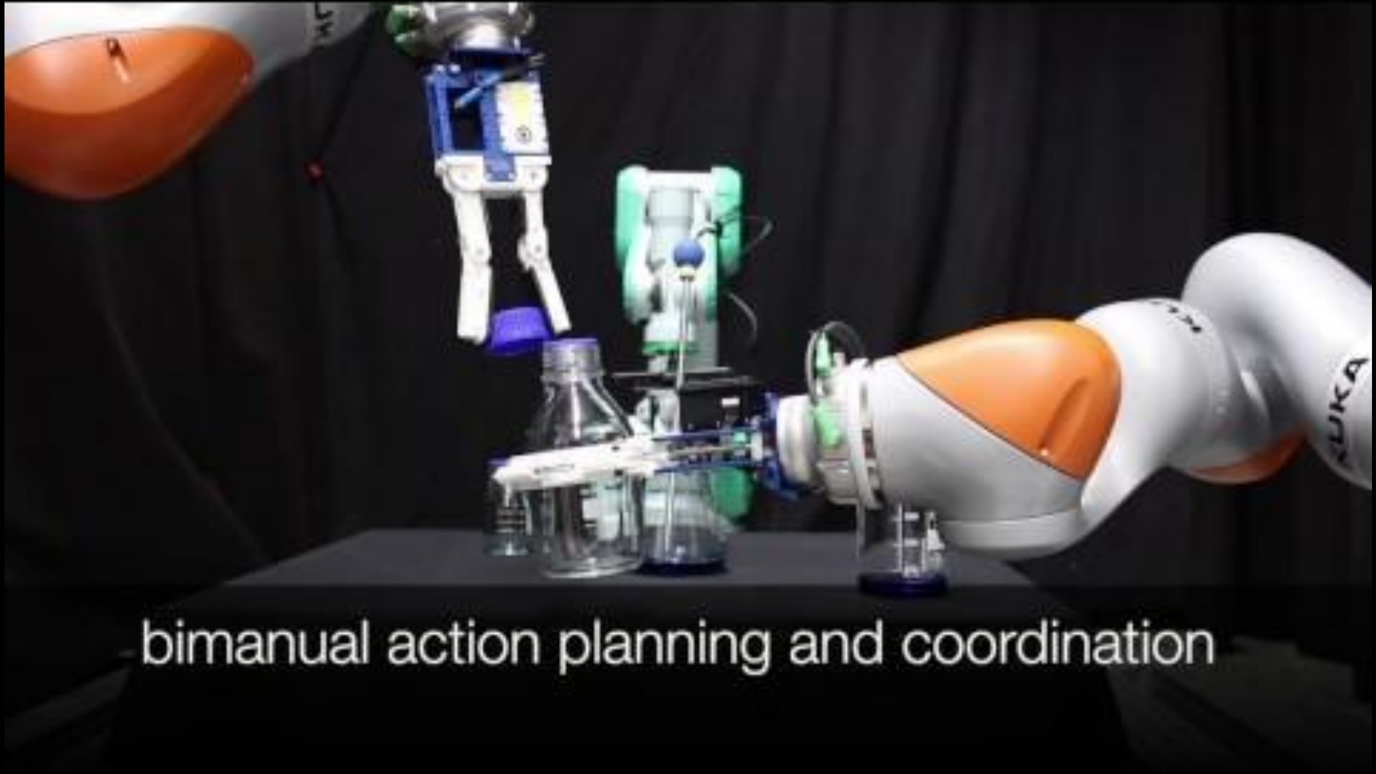
Despite **robot** vision is richer and more precise, robot motion is faster and more accurate their motor skills are inferior, **why?**

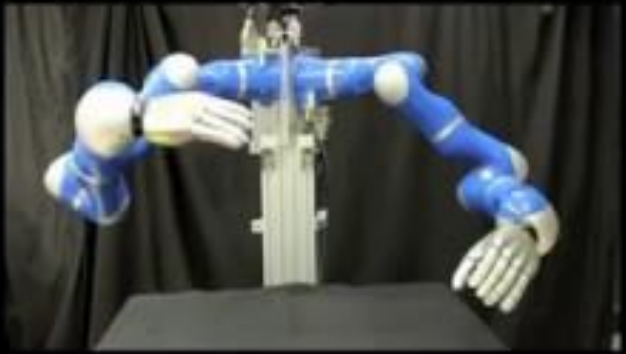
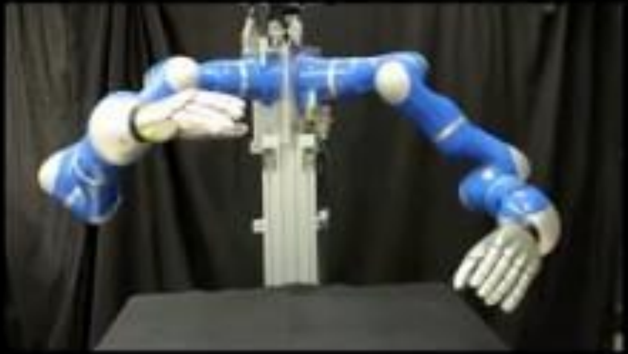
Why probabilistic methods?

- Uncertainties in the sensor measurements.
- Delays and transmission errors.
- Unmodeled dynamics (friction dynamics, coriolis forces, etc.).
- Partial observability.

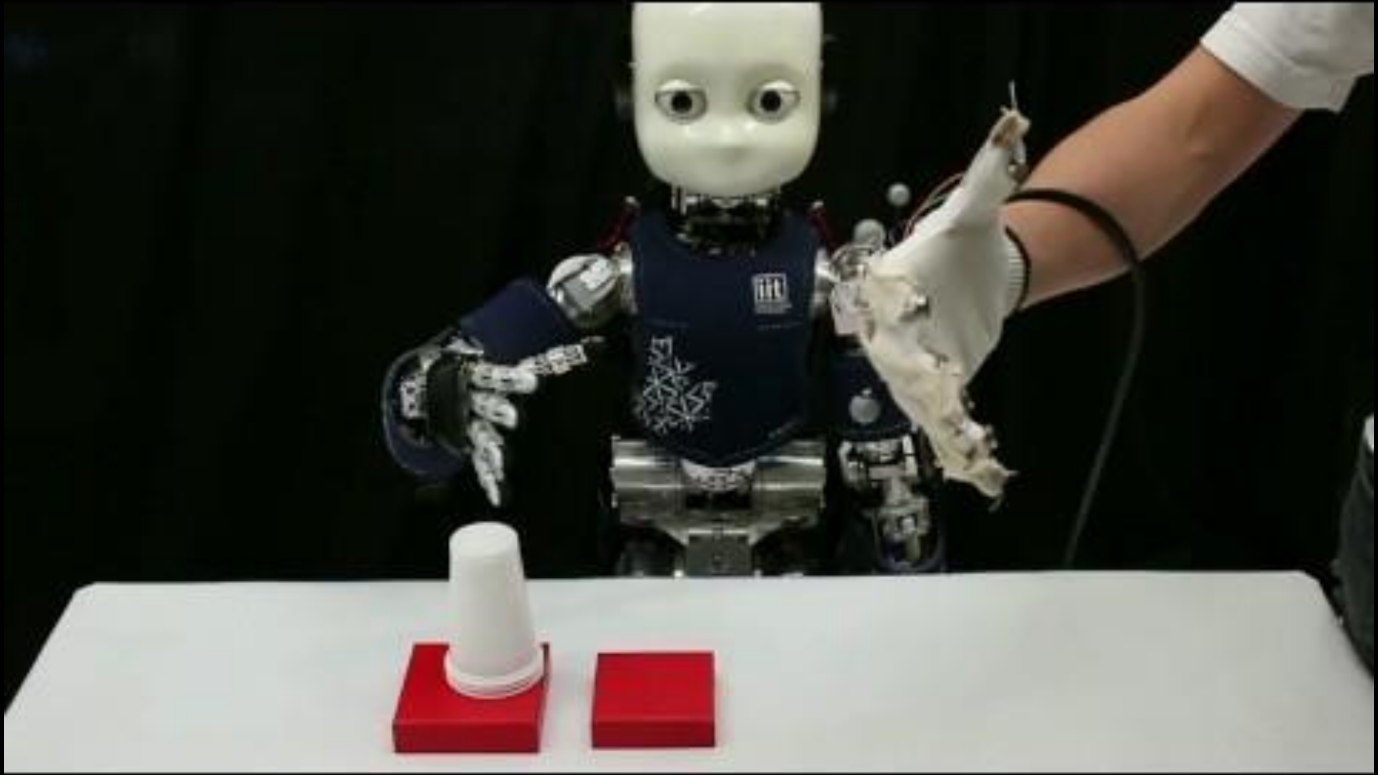
Why neural methods?

- The optimal methods structure / features are often uknown.
- Millions of data samples can be processed in O(n).
- Complex multimodal probability distributions can be represented (in contrast to commonly used unimodal Gassians).
- Predictions can be computed in realtime in O(1).


bimanual action planning and coordination


9:33:02 05/06/2015 UTC



Research questions

1. How can humans learn new motor skills within a few trials?
   a. "control only when necessary" - motor variability
   b. exploiting kinematic and task redundancy
   c. transfer of related skills

2. How do humans solve cognitive reasoning tasks in huge spaces?
   a. planning in stochastic environments
   b. inferring multiple solutions in milliseconds
   c. online model adaptation from intrinsic motivation signals.

Interested in a brief robotics history?

A brief historical review

Link to a more detailed history review

1920 **Karek Capek**: “robot” in his play “R.U.R.” (Rossum’s Universal Robots).

1941 **Isaac Asimov**: Three laws of “robotics”:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

A brief historical review

1968 “**Shakey**” of the “Stanford Research Institute” defines a landmark in robotics:

- basic planning and navigation skills.
- object detection and manipulation capabilities.

A brief historical review

1973 **Ichiro Kato** develops the first “full-scale” antrophomorphic humanoid, WABOT I.

A brief historical review

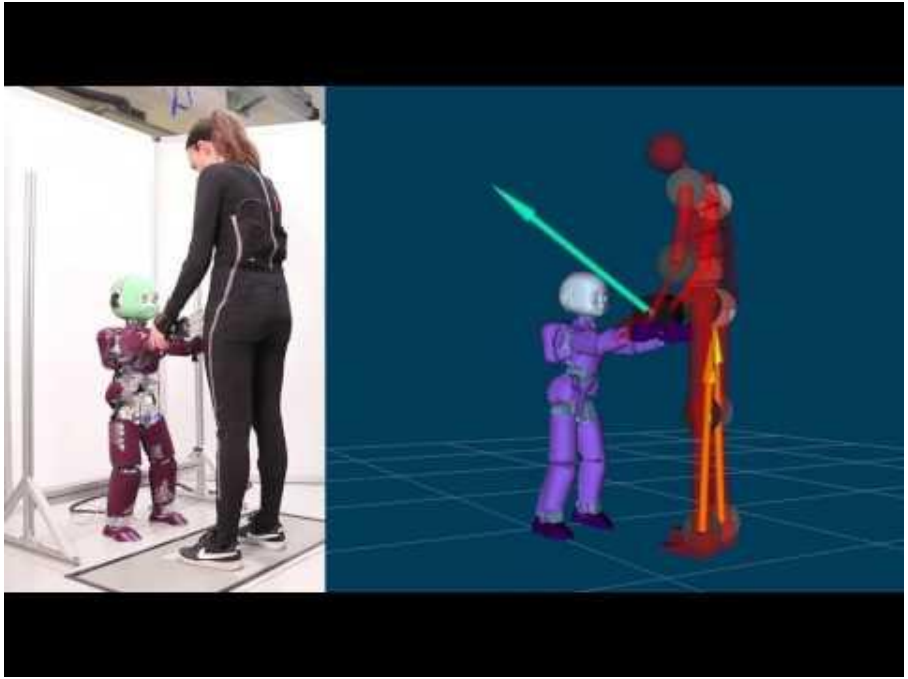
1996 **Honda** presents its P2

they started with E0 in 1986

the history of Hunda’s humanoids

A brief historical review

2004 The Italian Institute of Technologie presents the **ICub** (intelligent man-cub).

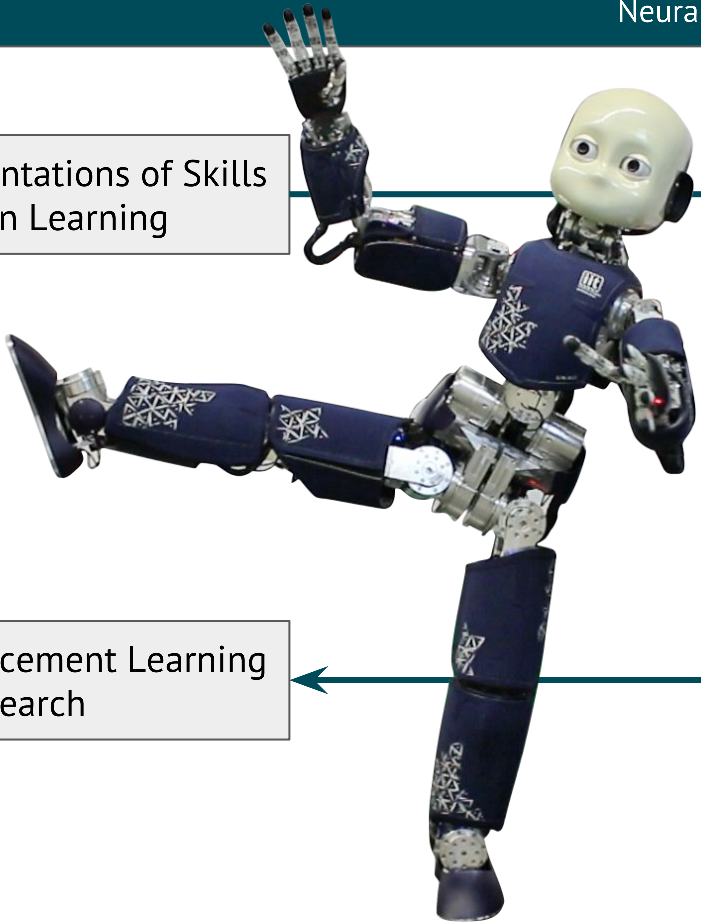


A brief historical review

2017 Boston dynamics’ **Atlas** impresses the robotics community.

II. Representations of Skills & Imitation Learning

V. Cognitive Reasoning & Planning



I. Kinematics, Dynamics & Model Learning

III. Feedback, Priorities & Torque Control

IV. Reinforcement Learning & Policy Search

VI. Sensor Integration & Fusion

I. Kinematics, Dynamics & Model Learning

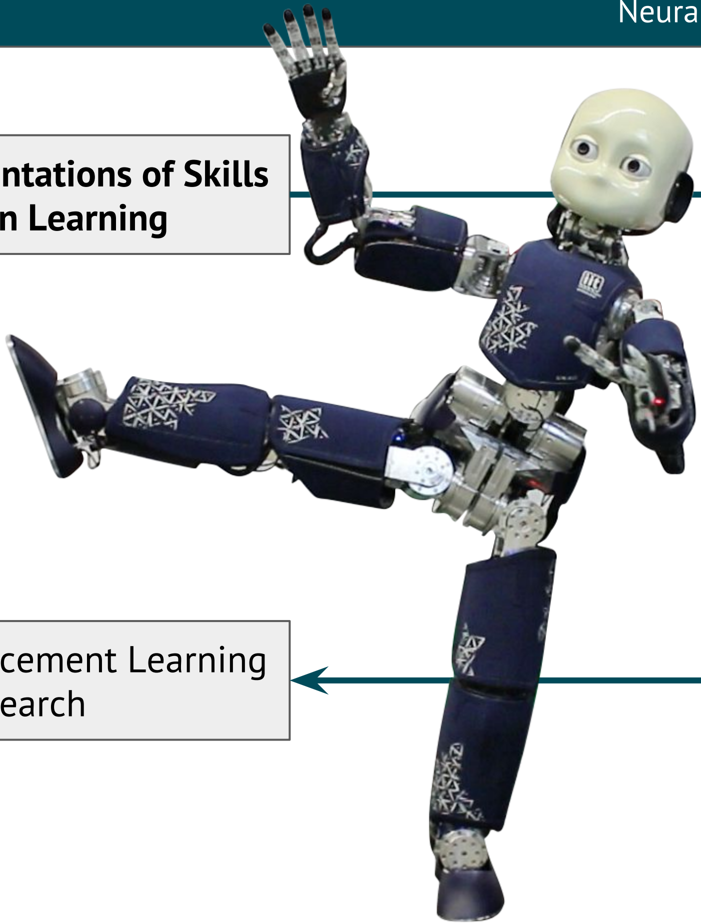
II. Representations of Skills & Imitation Learning

V. Cognitive Reasoning & Planning

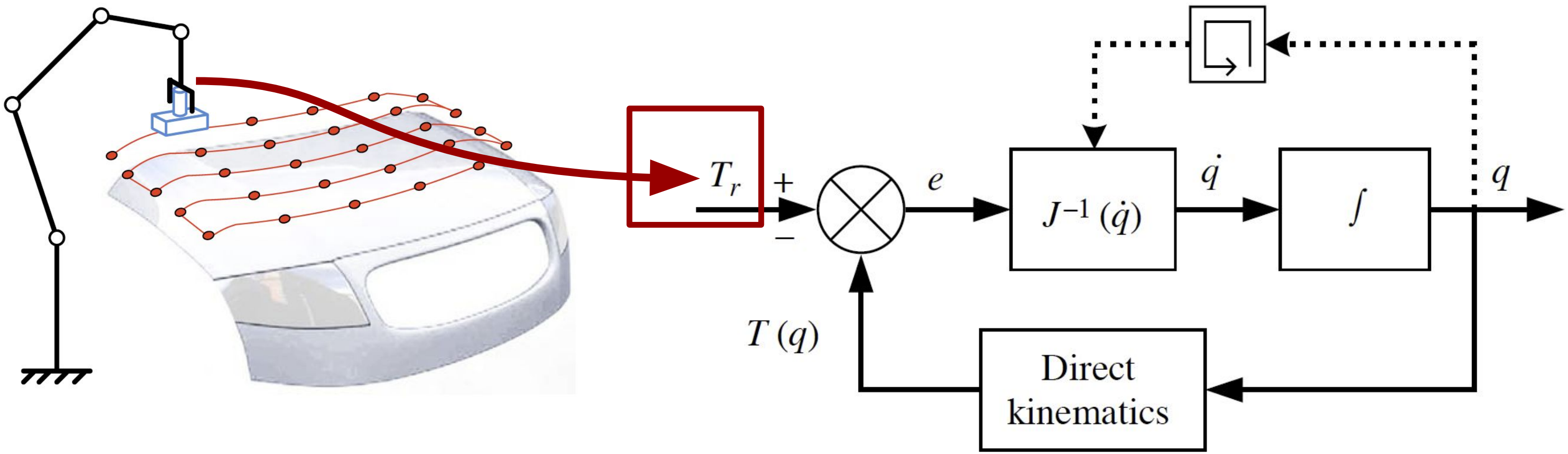
III. Feedback, Priorities & Torque Control

VI. Sensor Integration & Fusion

IV. Reinforcement Learning & Policy Search



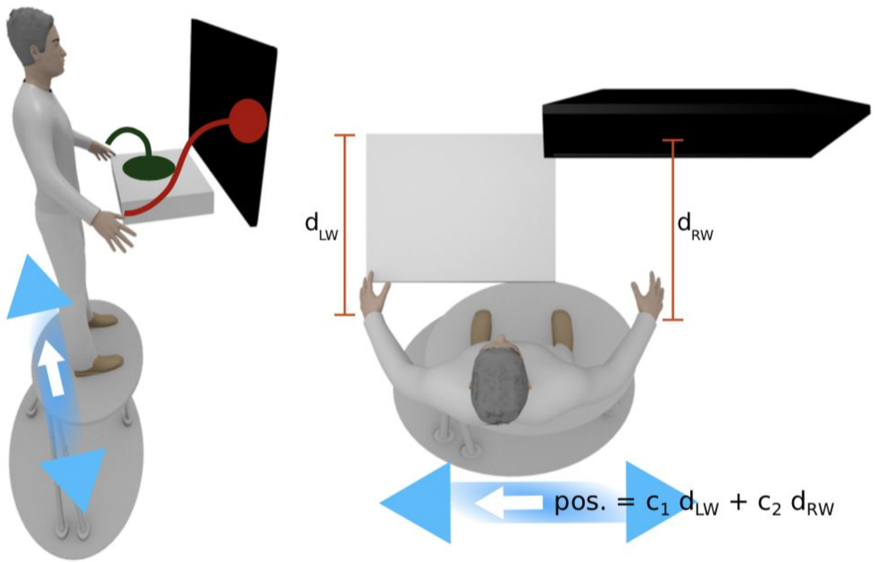
Where do we need representations of skills?

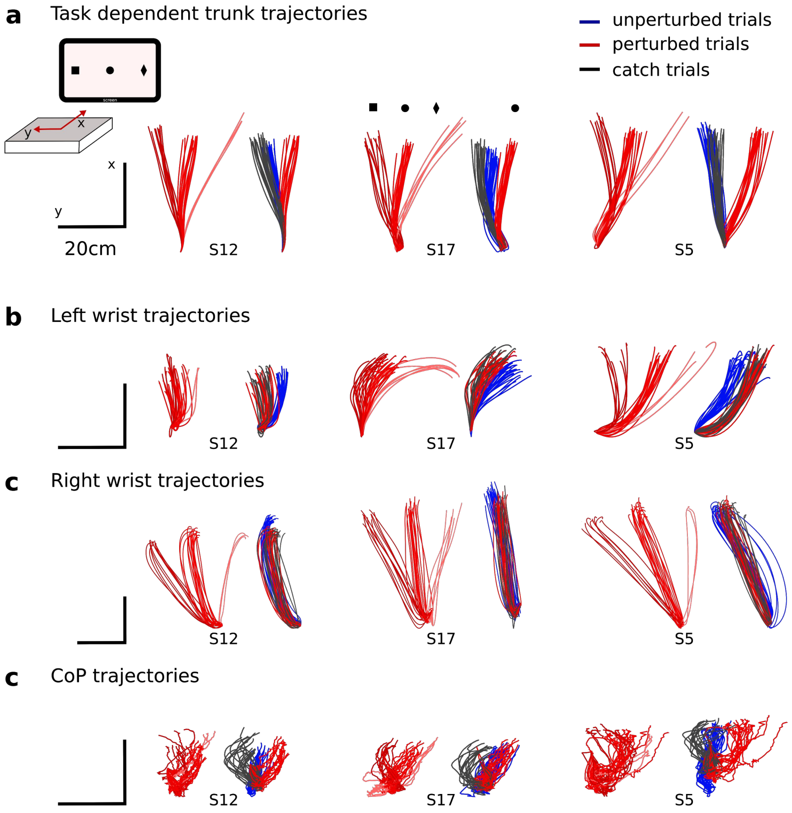


Fan Zeng, Beshah Ayalew and Mohammed Omar: Roboticc automotive paint curing using thermal signature feedback, 2009

II.1 Movement primitives.

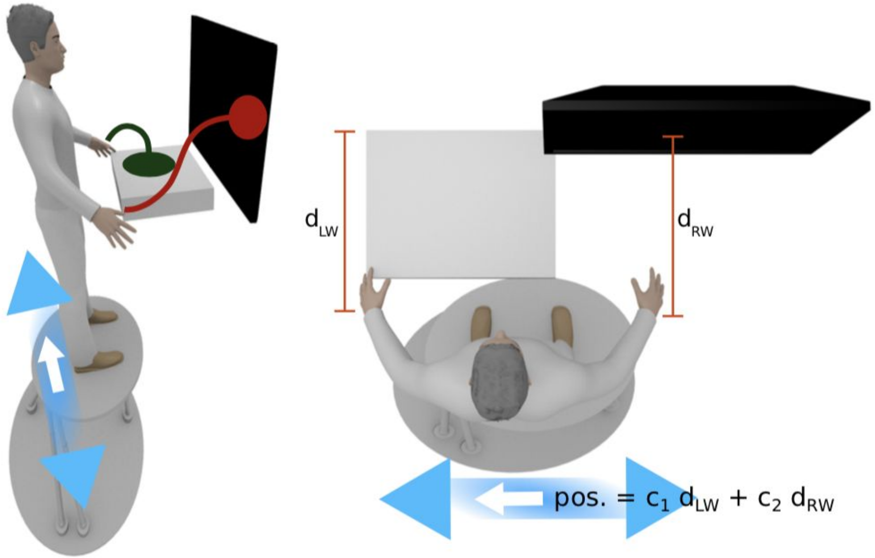
The complexity of skill representations





Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. [Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control.](#) Nature Publishing Group: Scientific Reports, 6 (28455), 2016.

The complexity of skill representations



Data:

- 17 markers with x,y,z at 100Hz
- 2 force plates at 100Hz (CoM at x,y)
- 9600 trials of 20 subjects of

- On avg. 100 samples per trial

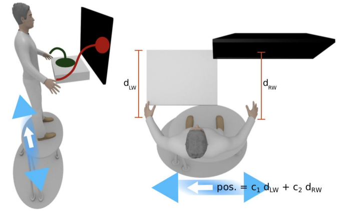
(17·3+2·2)·9600·100 **> 50 Mio. data pts**

Just for a single movement skill!

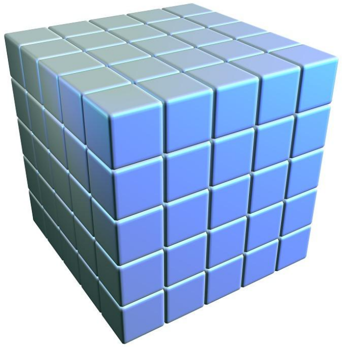
Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. [Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control.] Nature Publishing Group: Scientific Reports, 6 (28455), 2016.

Naive vector/matrix representation

scales in **O(d x T x K),** where d … number of joints, force plates or markers,
T … number of time steps per trial k = 1…K



50 Mio. data pts stored with 64 bits per double **> 3 GByte for movements of 1 second!**

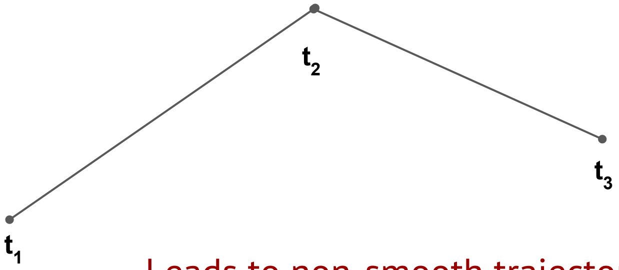


Even when we average over all 9600 trials we would need to store **5500 data points per second!**

Naive via-point representation

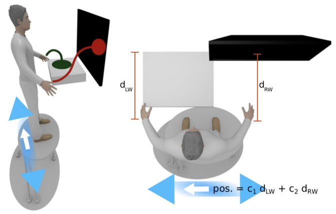
scales in **O(d x N x K),** where d … number of joints, force plates or markers,
n … number of via-points comp. from the avg. over K trials

$$f_i(t) = c_{i0} + c_{i1}(t - t_i) \quad t \in [t_i, t_{i+1}]$$



Leads to non-smooth trajectories!

Can we do better?
Yes by using the **dynamics model** for planning a route through via-points!



-3 via-pts per marker

-Averaging the via-pts over the 9600 trials

(17·3+2·2)·3 = **165 parameters to learn**

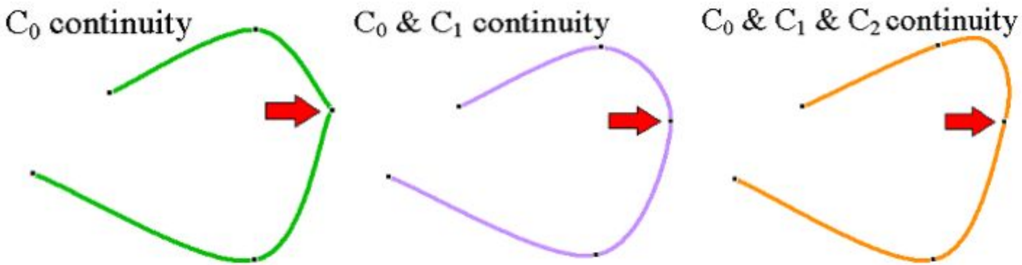
for a 55-dimensonal movement representation in ~10KB memory

Spline representation

Splines are piecewise polynomials (don't use solely polynomials)

$$f_i(t) = c_{i0} + c_{i1}(t - t_i) + c_{i2}(t - t_i)^2 + \cdots + c_{ik}(t - t_i)^k$$

$$t \in [t_i, t_{i+1}]$$



$$f_i'(t_{i+1}) = f_{i+1}'(t_{i+1}) \qquad f_i''(t_{i+1}) = f_{i+1}''(t_{i+1})$$

More slides on splines by Jernej Barbic, USC

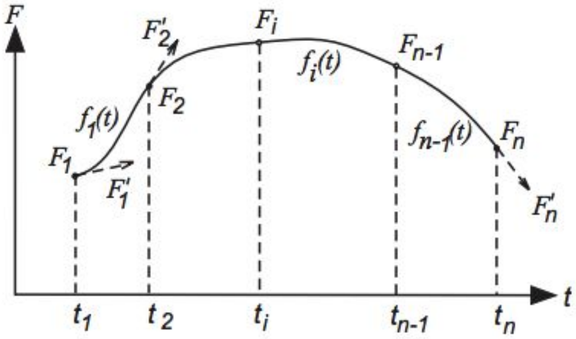


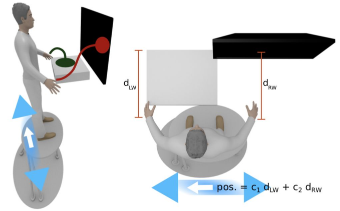
Fig. 1. A trajectory passing through n knots

Yisheng Guan, Kazuhito Yokoi, Olivier Stasse, Abderrahmane Kheddar. On Robotic Trajectory Planning Using Polynomial Interpolations. In Proceedings of the International Conference on Robotics and Biomimetics, 2005.

Spline representation

Scale in **O(d x n x *k*)**, where d … number of joints, force plates or markers, **n … number of knots** at times $t_1, t_2, \ldots, t_n$ of order *k*

$$f_i(t) = c_{i0} + c_{i1}(t - t_i) + c_{i2}(t - t_i)^2 + \cdots + c_{ik}(t - t_i)^k$$



-3 via-pts per marker

-Averaging the via-pts over the 9600 trials

-k=3 for cubic-splines (c_{i0}, c_{i1}, c_{i2}, c_{i3})

(17·3+2·2)·3·4 = **660**

parameters to learn!

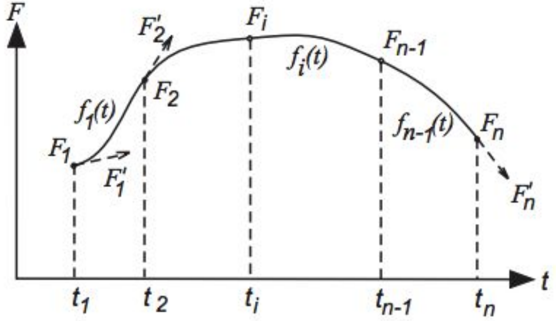
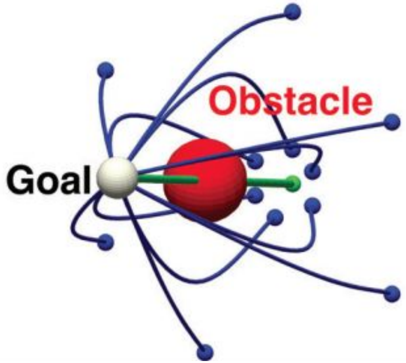


Fig. 1. A trajectory passing through n knots

Yisheng Guan, Kazuhito Yokoi, Olivier Stasse, Abderrahmane Kheddar. On Robotic Trajectory Planning Using Polynomial Interpolations. In Proceedings of the International Conference on Robotics and Biomimetics, 2005.

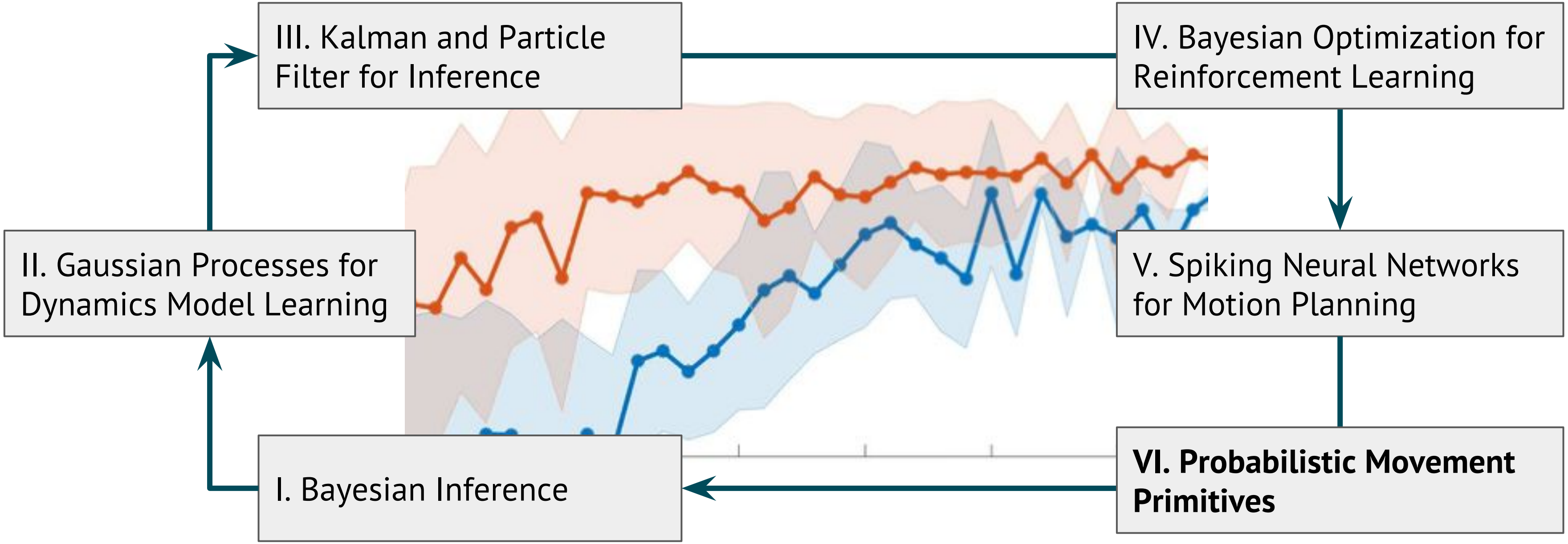
Desired features of skill representations

- **Compact** (few parameters to learn).
- **Smooth** (need to compute derivatives for velocities and controls).
- **Flexible** generalizables to different tasks (goal locations, orientations, etc.).
- Can be learnt from the data through **imitation learning (IM)**.
- Self-improvement through **reinforcement learning (RL)**.
- **Composable** through sequencing and **co-activation.**
- **Stochastic**, can model the variance of the data.
- **Coupled**, can model the coupling of joints.

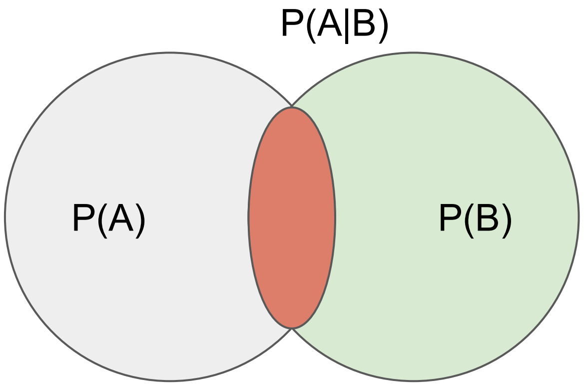


Ex. flexibility to start at different poses.

Probabilistic Methods for Robotics



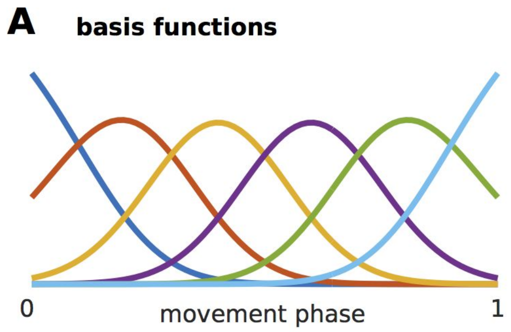
My approach: learning probabilistic models



Learning problem:

$P(A|B) = P(A, B)/P(B)$

given data samples from $P(A, B)$
assuming priors $P(A)$, $P(B)$


A
basis functions
0
movement phase
1

[1] Generative Model: $\boldsymbol{y}_t = \boldsymbol{\Phi}_t \, \boldsymbol{w}$

[2] Gaussian Features: $\phi_{t,i} = \frac{1}{\mathscr{Z}} \exp\left(-\frac{1}{2h}(z(t) - c_i)^2\right)$,

[3] Learning the Prior: $\boldsymbol{w}^{[i]} = \left(\boldsymbol{\Phi}_{1:T}^T \boldsymbol{\Phi}_{1:T} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}_{1:T}^T \boldsymbol{\tau}^{[i]}$.

[4] Model:

learned prior

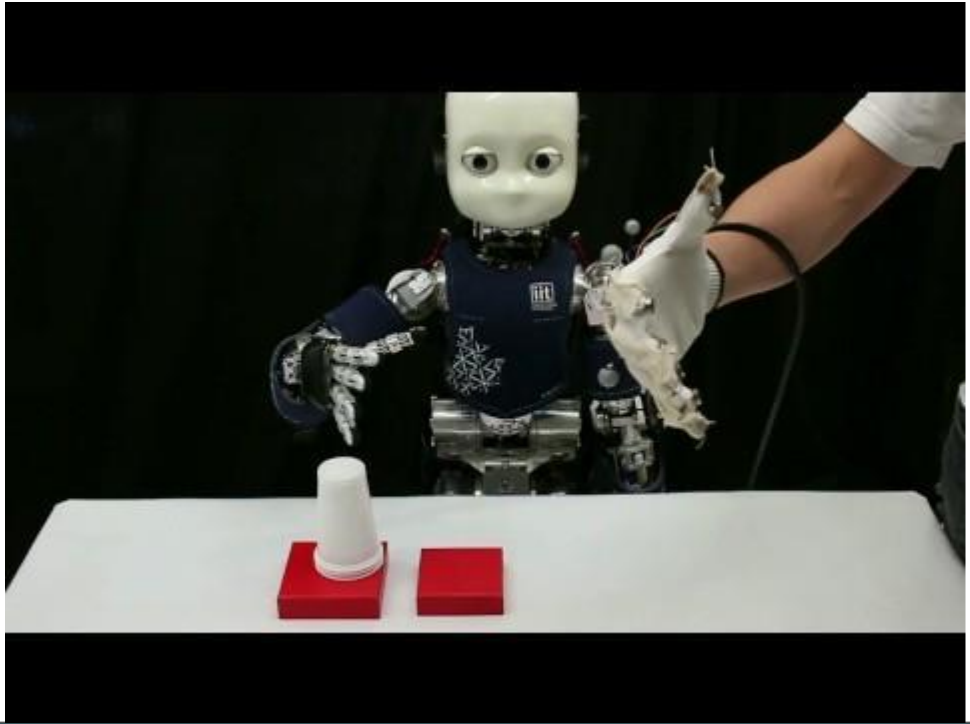
$$
\begin{aligned}
p(\boldsymbol{\tau}) &= \int p(\boldsymbol{\tau}|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w} \\
&= \int \mathcal{N}(\boldsymbol{y}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{w}, \boldsymbol{\Sigma}_y) \boxed{\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)} d\boldsymbol{w} \\
&= \mathcal{N}(\boldsymbol{y}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{w}, \; \boldsymbol{\Phi}_{1:T}\boldsymbol{\Sigma}_w\boldsymbol{\Phi}_{1:T}^T + \boldsymbol{\Sigma}_y) \; .
\end{aligned}
$$

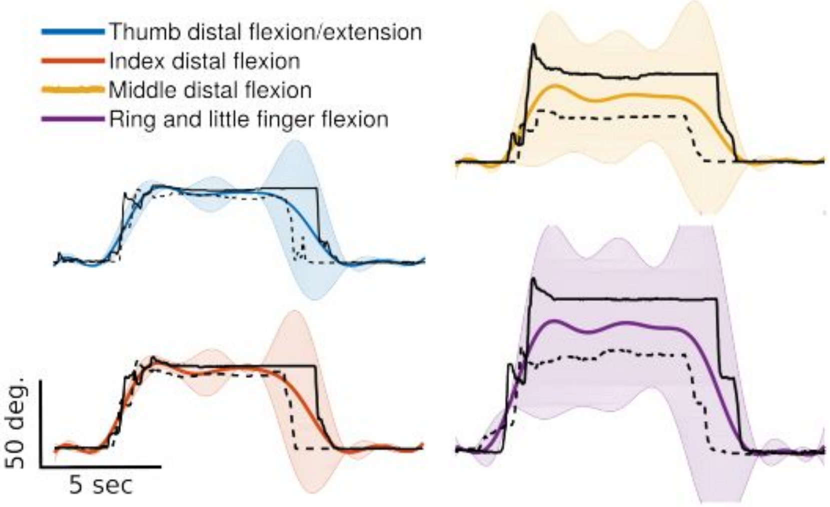
[5] Conditioning, given the prior $\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$

$$\begin{aligned} p(\boldsymbol{w}_o|\boldsymbol{o}) &\propto \mathcal{N}(\boldsymbol{o}|\boldsymbol{\Phi}_{\boldsymbol{o}}\boldsymbol{w}_o, \boldsymbol{\Sigma}_{\boldsymbol{o}})p(\boldsymbol{w}) \\ &:= \mathcal{N}\left(\boldsymbol{w}_o|\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{o}}, \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{o}}\right), \\ \text{with} \quad \boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{o}} &= \boldsymbol{\mu}_{\boldsymbol{w}} + \boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{o}}^T(\boldsymbol{\Sigma}_o + \boldsymbol{\Phi}_{\boldsymbol{o}}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{o}}^T)^{-1}(\boldsymbol{o} - \boldsymbol{\Phi}_o\boldsymbol{\mu}_{\boldsymbol{w}}), \\ \text{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{o}} &= \boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{o}}^T(\boldsymbol{\Sigma}_o + \boldsymbol{\Phi}_{\boldsymbol{o}}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}_{\boldsymbol{o}}^T)^{-1}\boldsymbol{\Phi}_{\boldsymbol{o}}\boldsymbol{\Sigma}_{\boldsymbol{w}}, \end{aligned}$$

Result: $$p(\tilde{\tau}) = \mathcal{N}\left(\tilde{\boldsymbol{y}}_{1:T}|\boldsymbol{\Phi}_{1:T}\boldsymbol{\mu}_{\boldsymbol{w}|\boldsymbol{o}}, \boldsymbol{\Phi}_{1:T}\boldsymbol{\Sigma}_{\boldsymbol{w}|\boldsymbol{o}}\boldsymbol{\Phi}_{1:T}^T + \boldsymbol{\Sigma}_y\right)$$

Movement model learning example





Rueckert, Elmar; Lioutikov, Rudolf; Calandra, Roberto; Schmidt, Marius; Beckerle, Philipp; Peters, Jan. [Low-cost Sensor Glove with Force Feedback for Learning from Demonstrations using Probabilistic Trajectory Representations.]() ICRA 2015 Workshop on Tactile and force sensing for autonomous compliant intelligent robots, 2015.

How do we train the model from data?

$$\mathbf{q}_t = [q_t^{[1]}, q_t^{[2]}, \ldots, q_t^{[d]}]^T$$
$$\forall t \in \mathbb{N}_0$$

- Kinesthetic teaching (see the picture).
- Teleoperation (e.g., by using a joystick).
- Visual observation (using cameras or optical markers).
- Sensor suits (IMUs, e.g., Xsense.com).

The last two approaches require to map the data onto the robot which is often problematic!

Imitation learning

Given: $\mathbf{q}_t = [q_t^{[1]}, q_t^{[2]}, \dots, q_t^{[d]}]^T$

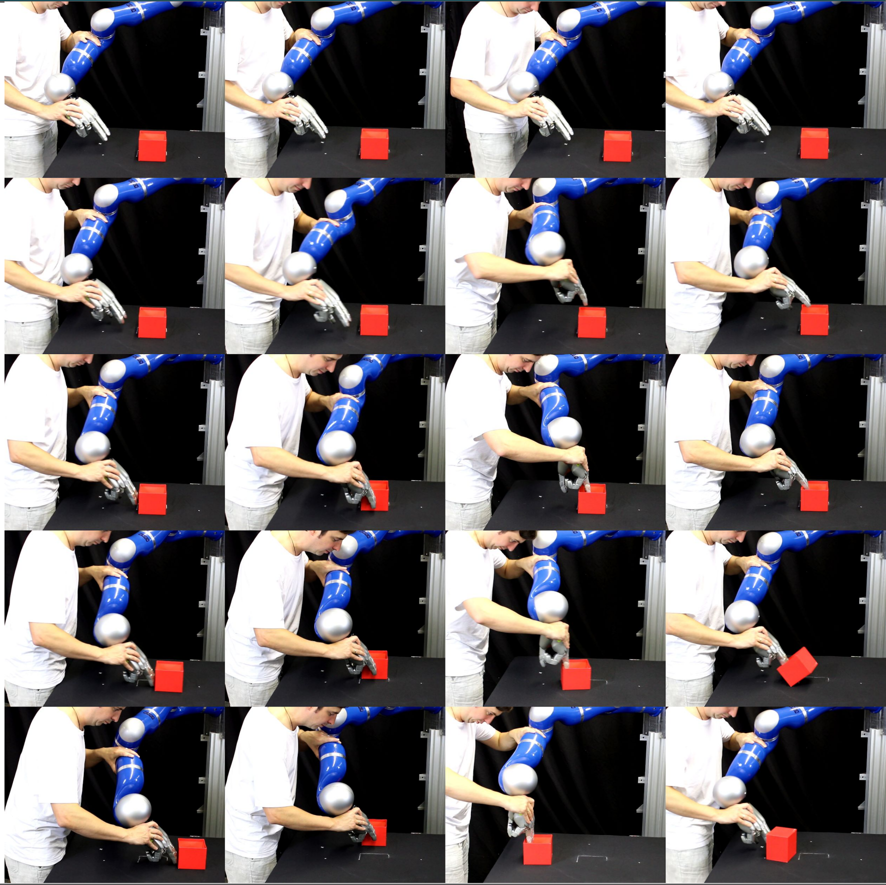
$\forall t \in \mathbb{N}_0$

Or in vector notation per dim. *d*:

$$\mathbf{q}^{[d]} = [q_1^{[d]}, q_2^{[d]}, \dots, q_T^{[d]}]^T$$

Let’s consider only one dimension:

$$\mathbf{q} = [q_1, q_2, \dots, q_T]^T$$

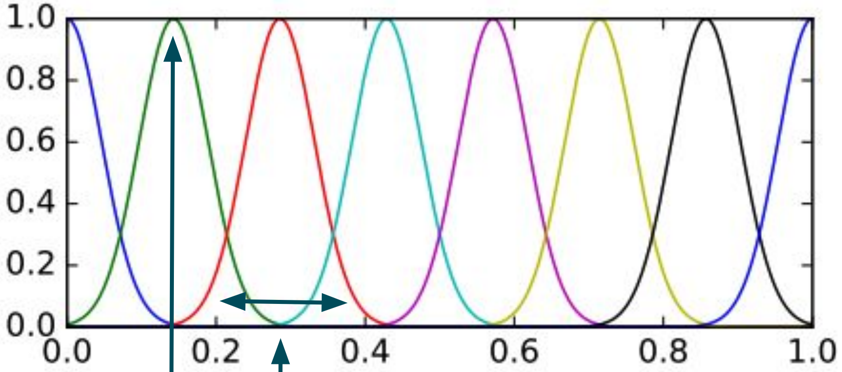


Radial basis functions as features

Modeling complex shapes through Gaussians

$$f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t) w_j}{\sum_{j=1}^{N} \Psi_j(t)}$$

$$\Psi_j(t) = \exp(-1/(2\sigma^2)(x(t) - c_j)^2)$$



Note N=8 Gaussian basis functions are used here

Radial basis functions as features

Modeling complex shapes through Gaussians

fixed basis functions

scaled by learnable parameters

$$f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t) w_j}{\sum_{j=1}^{N} \Psi_j(t)}$$

normalization

Imitation learning

I. Compute the **target** function from the data:

$$\tilde{\mathbf{f}} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$$

II. Compute the **model's** function term:

$$\mathbf{f} = \mathbf{\Psi}\mathbf{w}$$ from

$$f(t) = \frac{\sum_{j=1}^{N} \Psi_j(t) w_j}{\sum_{j=1}^{N} \Psi_j(t)}$$

where

$$\mathbf{\Psi} = \begin{bmatrix} \bar{\Psi}_1^{[1]}, & \bar{\Psi}_1^{[2]}, & \ldots, & \bar{\Psi}_1^{[N]} \\ \bar{\Psi}_2^{[1]}, & \ldots, & \ldots, & \ldots \\ \ldots & & & \\ \bar{\Psi}_T^{[1]}, & \ldots, & \ldots, & \bar{\Psi}_T^{[N]} \end{bmatrix}$$

$$\bar{\Psi}_t^{[j]} = \frac{\Psi_j(t)}{\sum_{j=1}^{N} \Psi_j(t)}$$

Imitation learning

I. Compute the **target** function from the data:

$$\tilde{\mathbf{f}} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$$

II. Compute the **model's** function term:

$$\mathbf{f} = \mathbf{\Psi}\mathbf{w}$$ from

III. Minimizing the objective:

$$J = \frac{1}{2}(\tilde{\mathbf{f}} - \mathbf{f})^T(\tilde{\mathbf{f}} - \mathbf{f}) = \frac{1}{2}(\tilde{\mathbf{f}} - \mathbf{\Psi}\mathbf{w})^T(\tilde{\mathbf{f}} - \mathbf{\Psi}\mathbf{w})$$

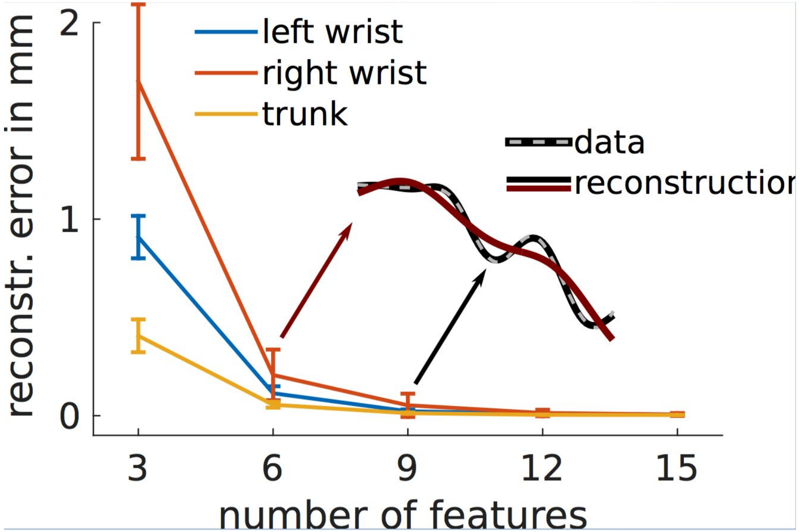
Results in:

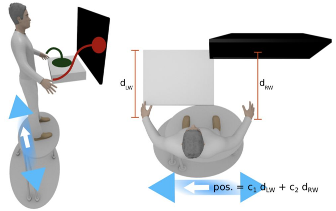
$$\boxed{\mathbf{w} = (\mathbf{\Psi}^T\mathbf{\Psi} + \lambda\mathbf{I})^{-1}\mathbf{\Psi}^T\tilde{\mathbf{f}}}$$

Link to a nice related tutorial

How many basis functions are optimal?

Depends on the task and has to be numerically evaluated!

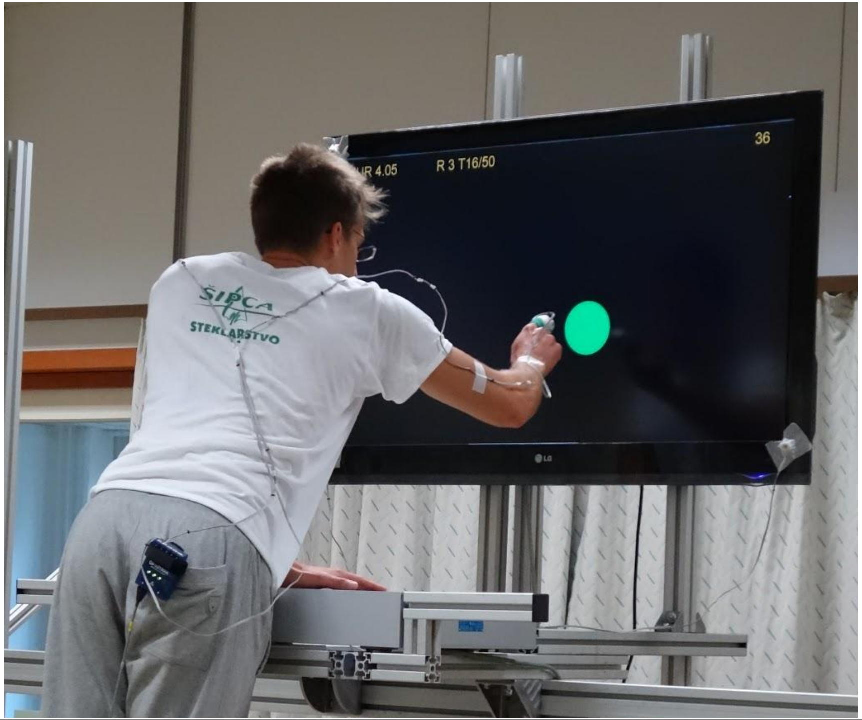


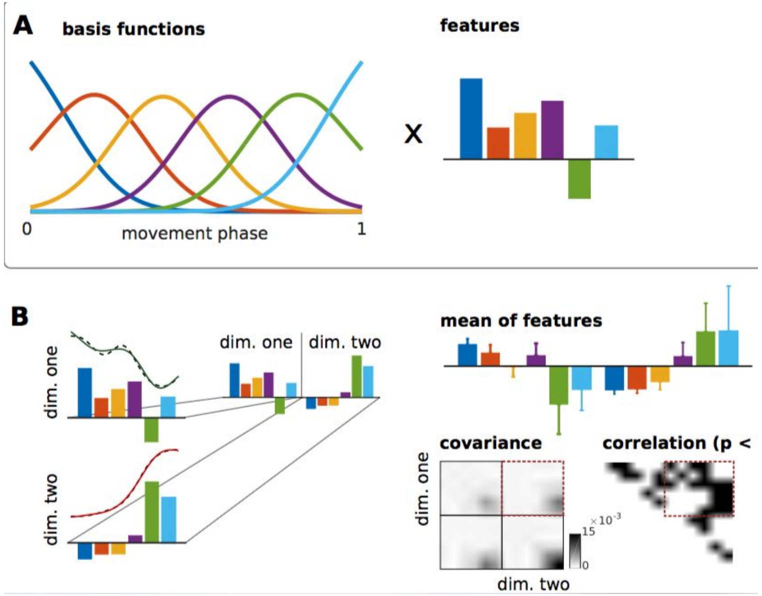


-6 Gaussians per dim.

- (17·3+2·2)·10 **= 550 parameters to learn** for a 55-dimensonal movement representation

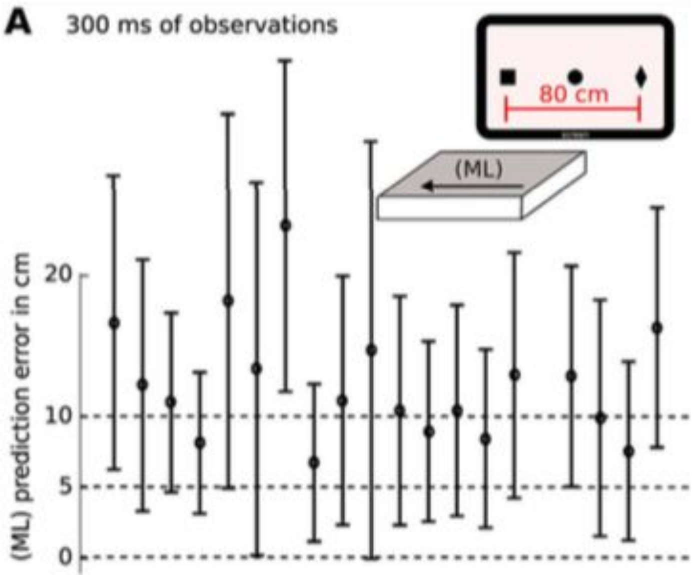
Imitation learning through optical markers





Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. [Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control.] Nature Publishing Group: Scientific Reports, 6 (28455), 2016.

Imitation learning through optical markers



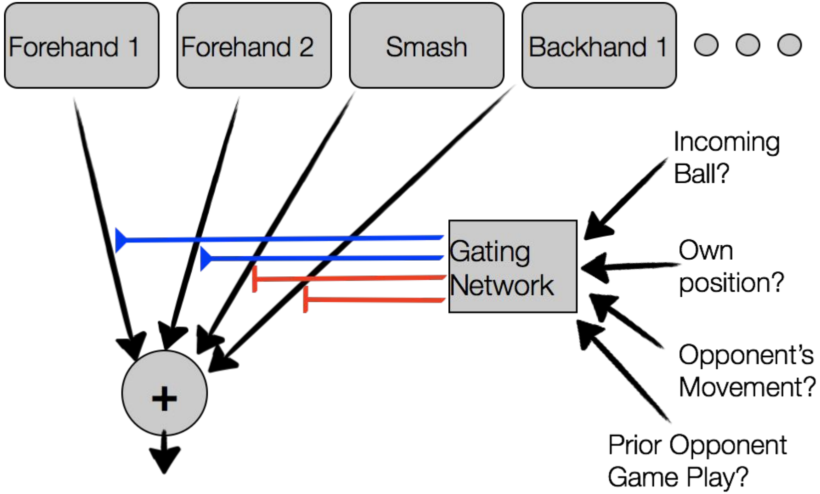
Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan. [Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control.] Nature Publishing Group: Scientific Reports, 6 (28455), 2016.

Is one movement primitive enough?

No!

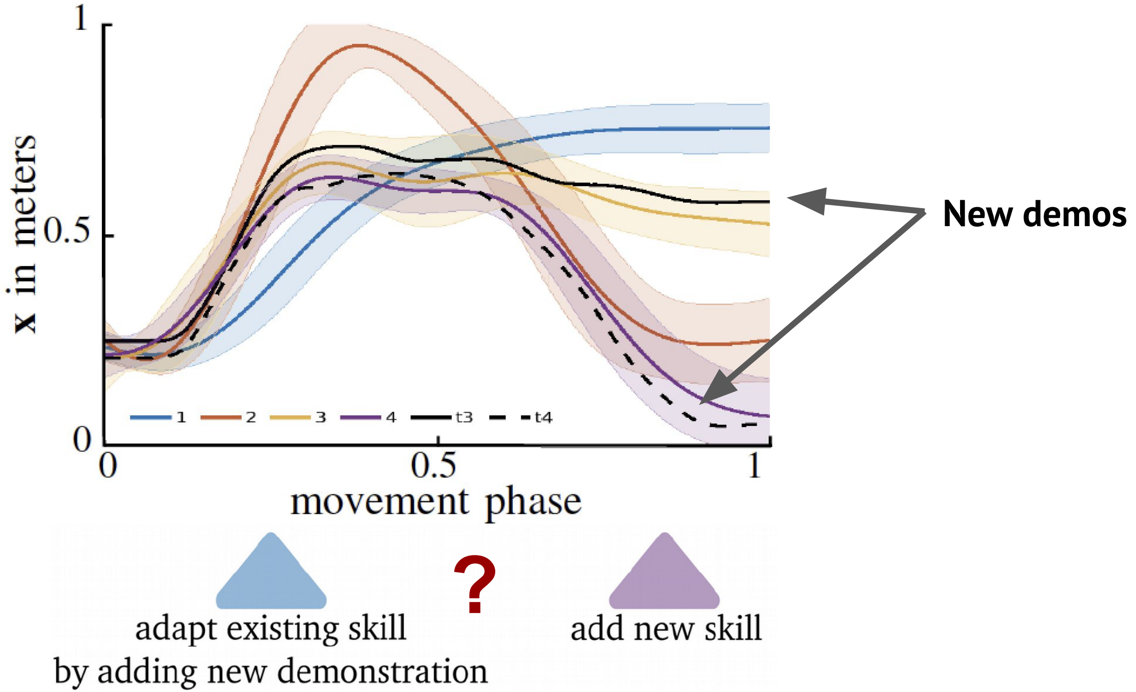
- Complex tasks require a large number of primitives.
- Reusable primitives can be sequenced or co-activated (in time).
- Non-homogeneous spaces require separate primitives (in space).
- Tradeoff between the number of primitives and their complexity (num. of Gaussians)!

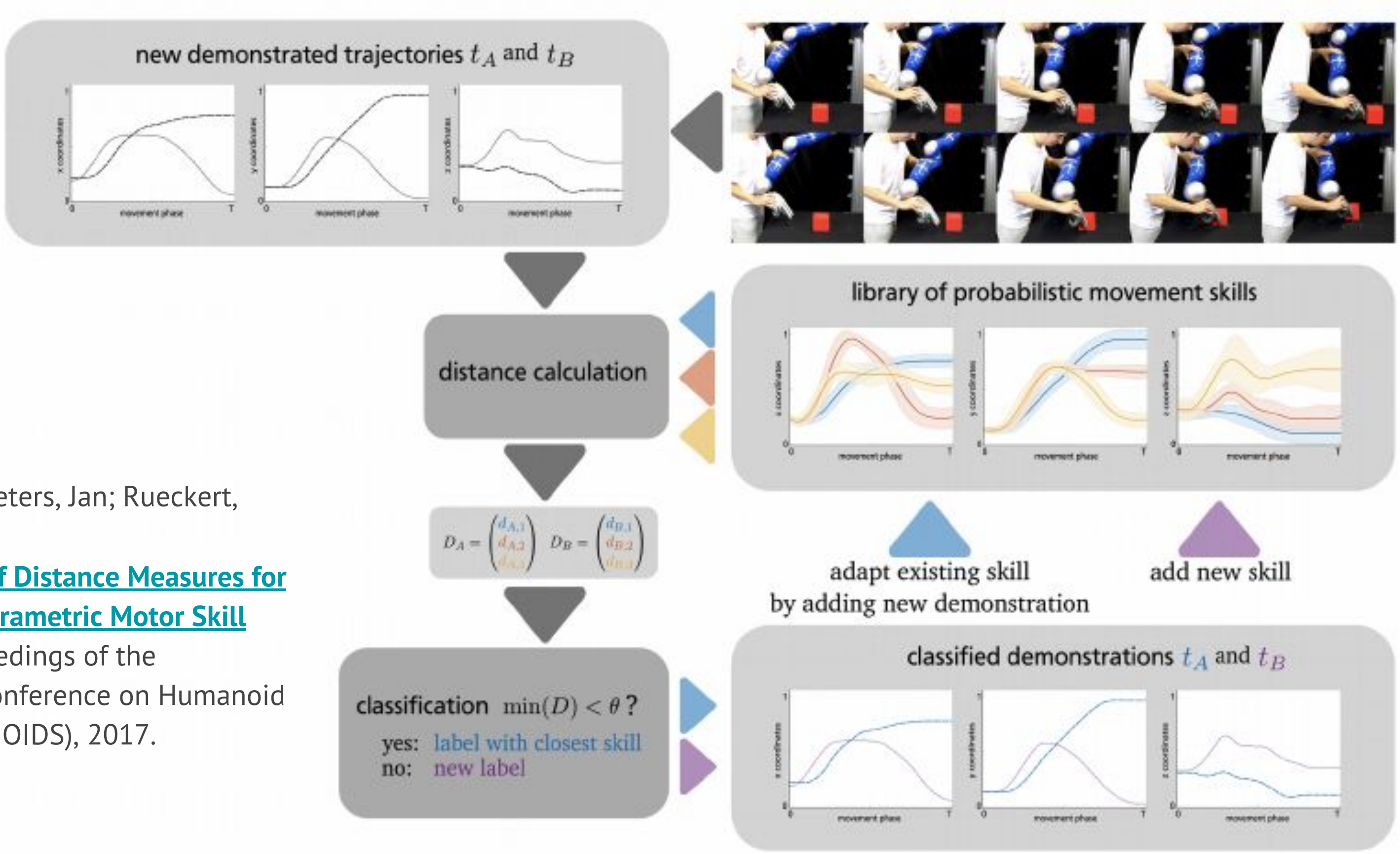
Imitation learning of a library of primitives



Muelling, K.; Kober, J.; Kroemer, O.; Peters, J. (2013). [Learning to Select and Generalize Striking Movements in Robot Table Tennis](), *International Journal of Robotics Research (IJRR)*, **32**, **3**, pp.263-279.

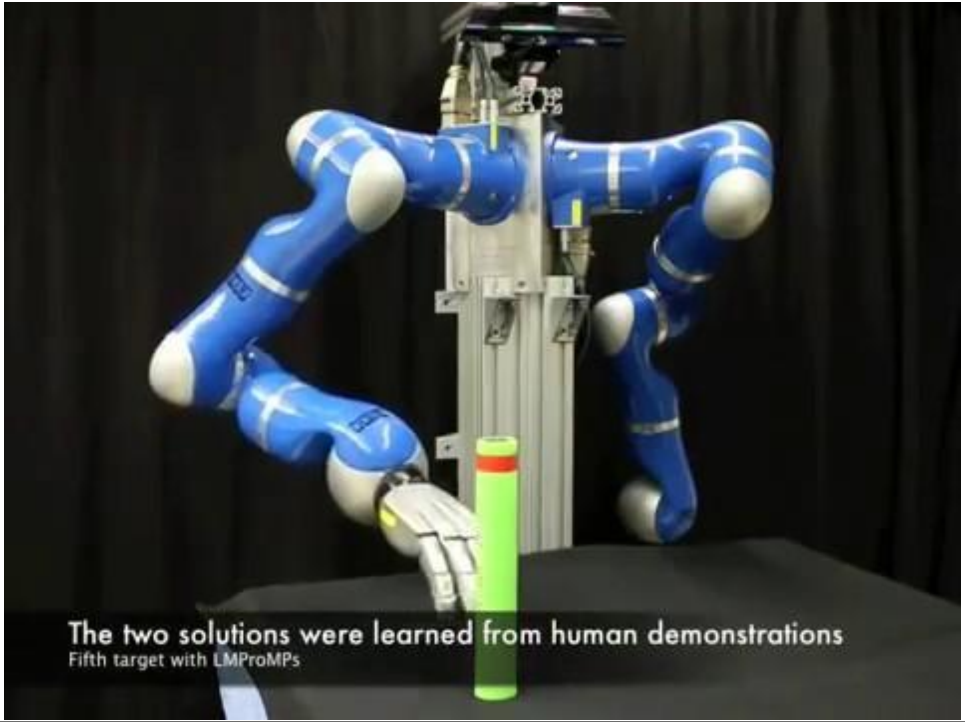
Incremental Imitation learning a primitive library

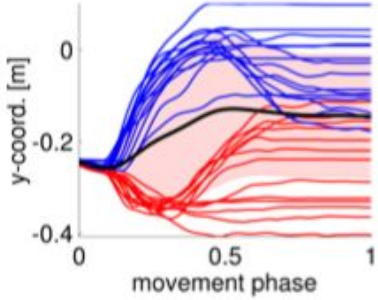


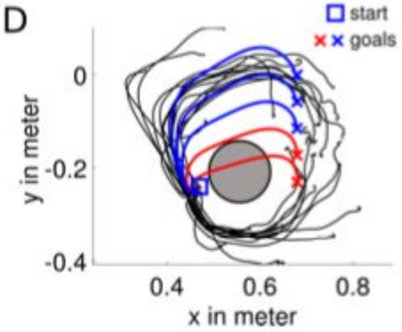


Stark, Svenja; Peters, Jan; Rueckert, Elmar.
[A Comparison of Distance Measures for Learning Nonparametric Motor Skill Libraries.]() Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), 2017.

When a single primitive is not sufficient



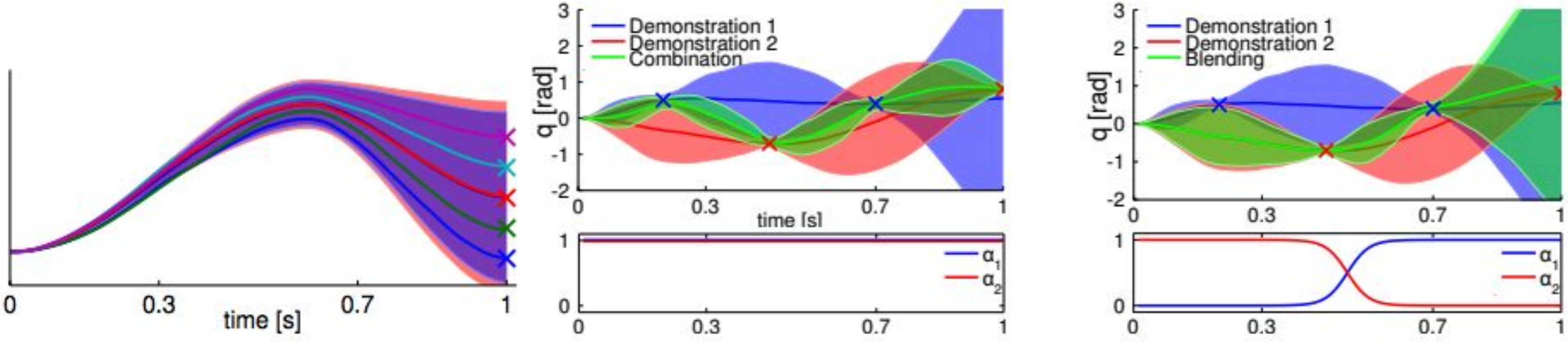




Rueckert, Elmar; Mundo, Jan; Paraschos, Alexandros; Peters, Jan; Neumann, Gerhard. [Extracting Low-Dimensional Control Variables for Movement Primitives.] Proceedings of the International Conference on Robotics and Automation (ICRA), 2015.

Can we generalize?

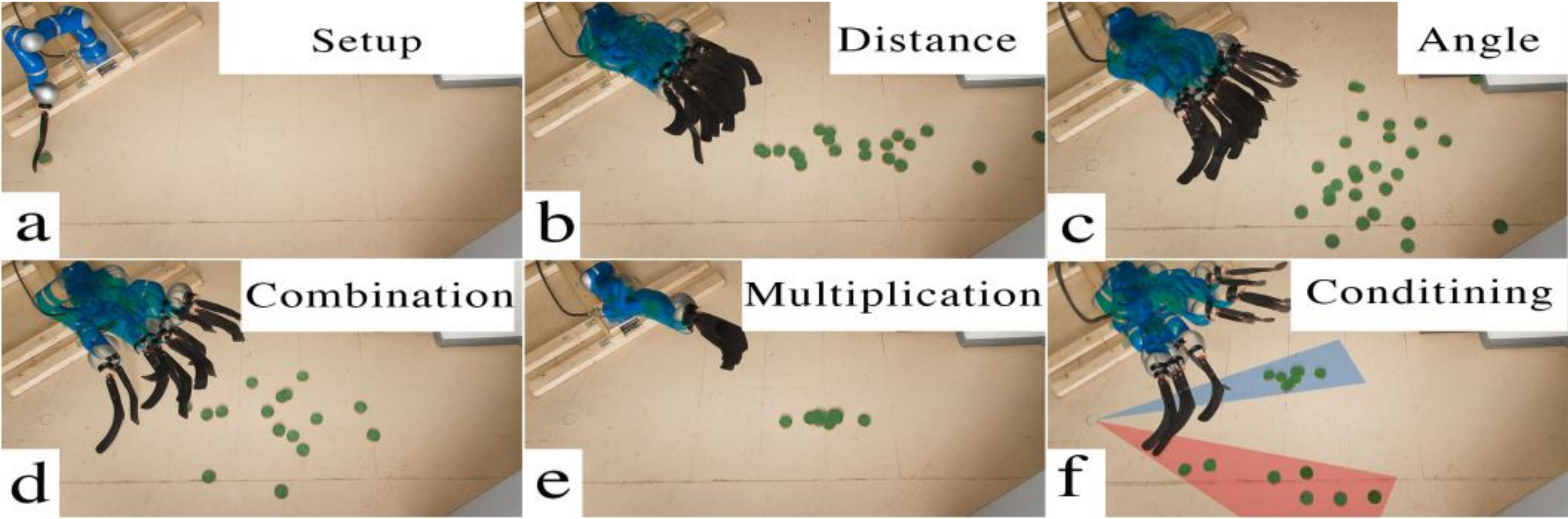
Using **probabilistic trajectory models** which are discussed in **Part Two**!



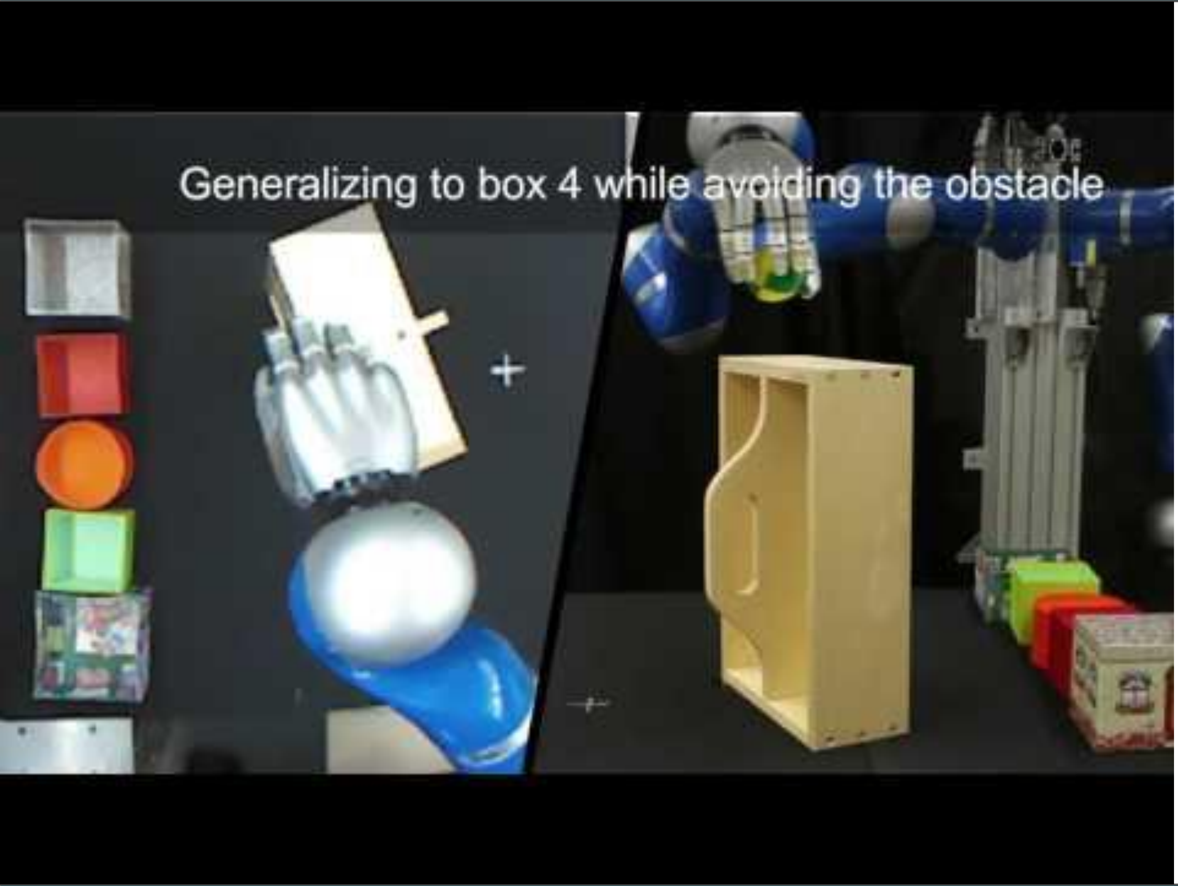
(a) Conditioning (b) Combination (c) Blending

Paraschos, Alexandros; Daniel, Christian; Peters, Jan; Neumann, Gerhard. Probabilistic Movement Primitives, *Advances in Neural Information Processing Systems (NIPS)*, MIT Press, 2013.

II.3 Example of probabilistic movement primitives.



Paraschos, Alexandros; Daniel, Christian; Peters, Jan; Neumann, Gerhard. Probabilistic Movement Primitives, *Advances in Neural Information Processing Systems (NIPS)*, MIT Press, 2013.



Koert, D.; Maeda, G.; Lioutikov, R.; Neumann, G. & Peters, J.
[Demonstration Based Trajectory Optimization for Generalizable Robot Motions.]
Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), 2016

Ewerton, M.; Neumann, G.; Lioutikov, R.; Ben Amor, H.; Peters, J.; Maeda, G. (2015). Learning Multiple Collaborative Tasks with a Mixture of Interaction Primitives, *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp.1535--1542.

You want to test PTMs yourself?

- https://rob.ai-lab.science/wp/resources/code/MATLAB_ProbabilisticTrajectoryModel_2016Rueckert.zip Matlab code.

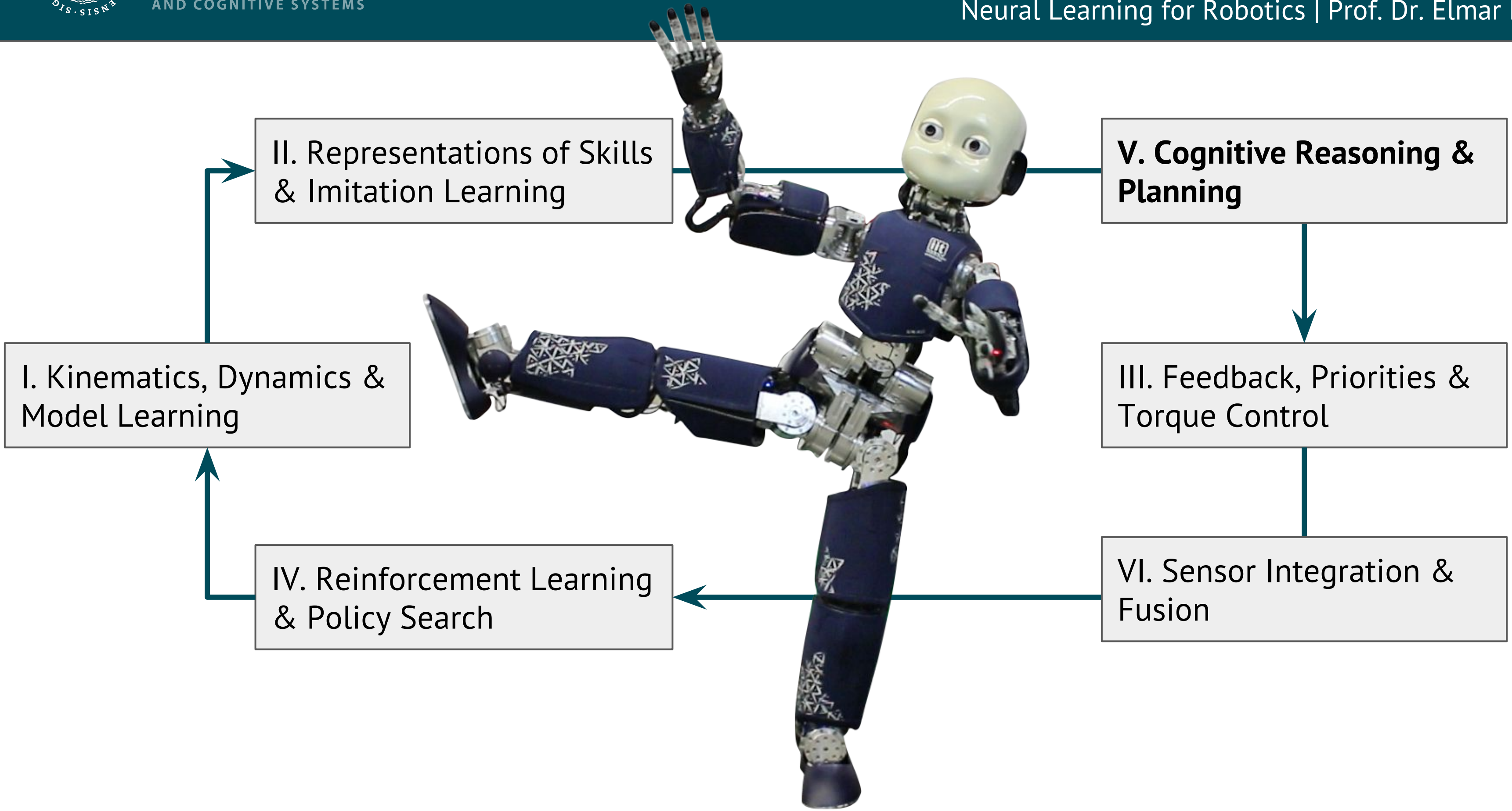
- More details and exercises in:

my online lectures at https://ai-lab.science more at: https://rob.ai-lab.science/publications/

Rueckert, Elmar; Camernik, Jernej; Peters, Jan; Babic, Jan
Probabilistic Movement Models Show that Postural Control Precedes and Predicts Volitional Motor Control Journal Article
Nature Publishing Group: Scientific Reports, 6 (28455), Impact Factor 4.122(‘17), 2016.

Rueckert, Elmar; Mundo, Jan; Paraschos, Alexandros; Peters, Jan; Neumann, Gerhard
Extracting Low-Dimensional Control Variables for Movement Primitives Inproceedings
Proceedings of the International Conference on Robotics and Automation (ICRA), 2015.

Rueckert, Elmar; Lioutikov, Rudolf; Calandra, Roberto; Schmidt, Marius; Beckerle, Philipp; Peters, Jan
Low-cost Sensor Glove with Force Feedback for Learning from Demonstrations using Probabilistic Trajectory Representations Inproceedings
ICRA 2015 Workshop on Tactile and force sensing for autonomous compliant intelligent robots, 2015.

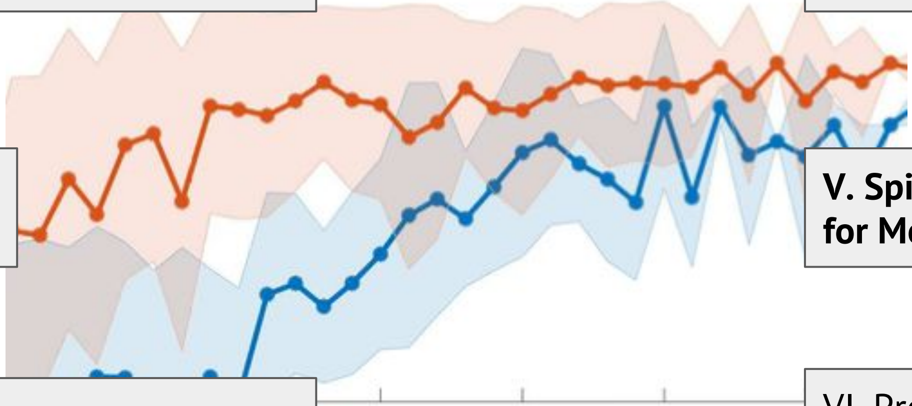

II. Representations of Skills & Imitation Learning
V. Cognitive Reasoning & Planning
I. Kinematics, Dynamics & Model Learning
III. Feedback, Priorities & Torque Control
IV. Reinforcement Learning & Policy Search
VI. Sensor Integration & Fusion

Choose your topic!

III. Kalman and Particle Filter for Inference

IV. Bayesian Optimization for Reinforcement Learning

II. Gaussian Processes for Dynamics Model Learning



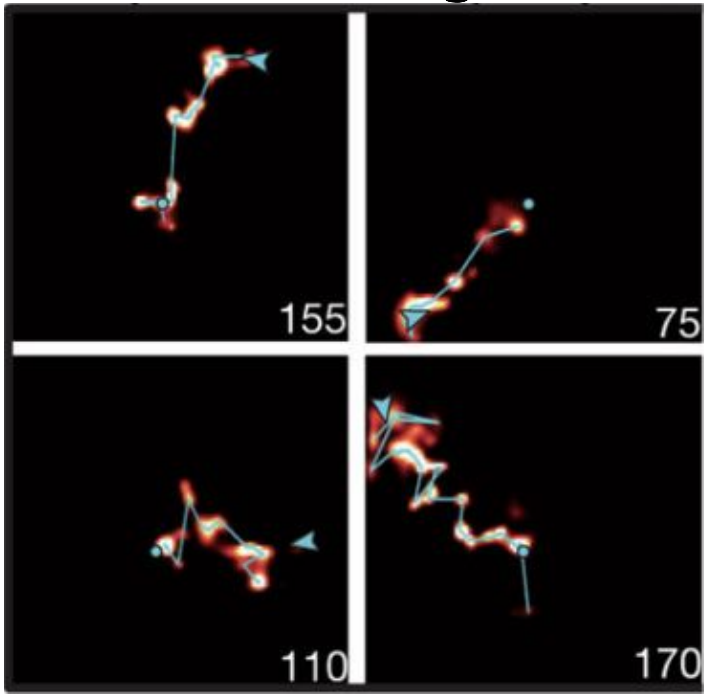
V. Spiking Neural Networks for Motion Planning

I. Bayesian Inference

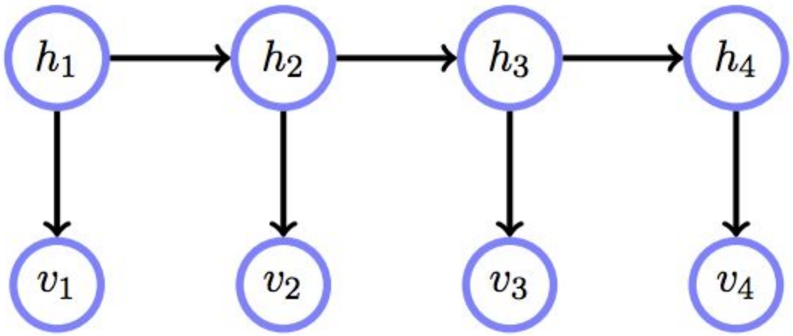
VI. Probabilistic Movement Primitives

Predictive models of rats' navigation skills

Predictive models of rats' navigation skills



Difference btw. Filtering, Smoothing and Predictions



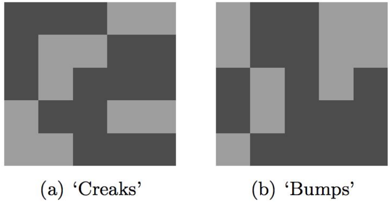
$$p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1)\prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1})$$

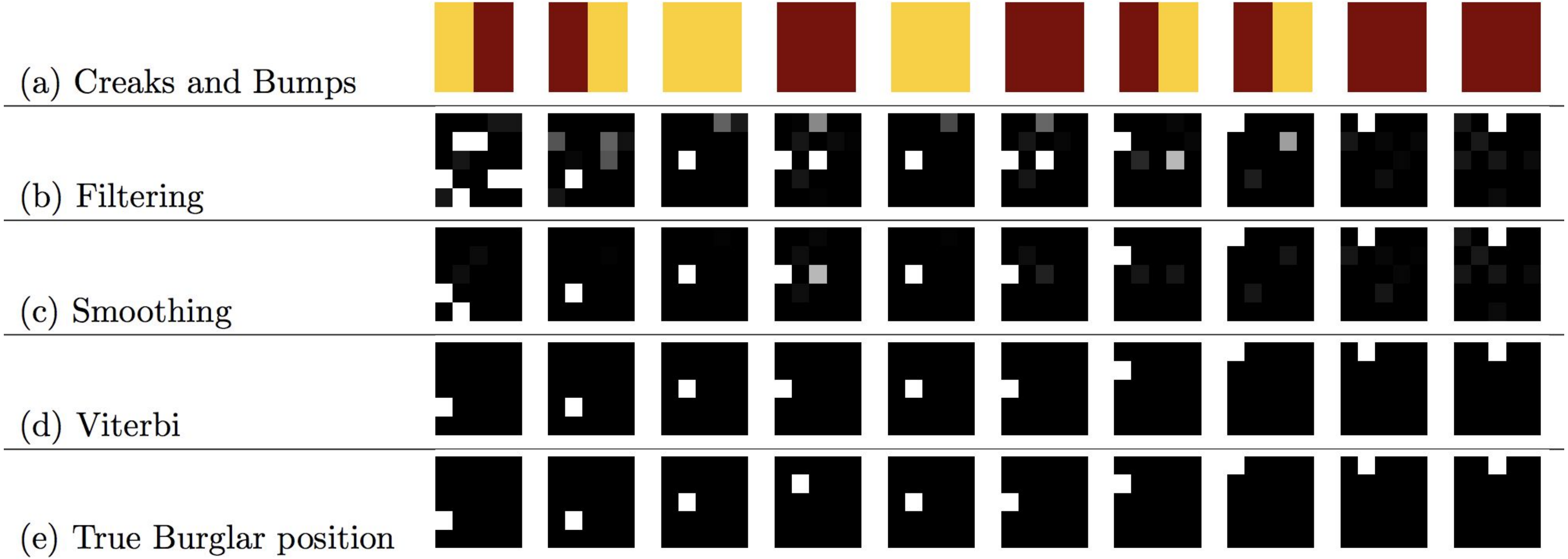
Difference btw. Filtering, Smoothing and Predictions

$$p(h_{1:T}, v_{1:T}) = p(v_1|h_1)p(h_1) \prod_{t=2}^{T} p(v_t|h_t)p(h_t|h_{t-1})$$

Filtering	(Inferring the present)	$p(h_t\|v_{1:t})$	
Prediction	(Inferring the future)	$p(h_t\|v_{1:s})$	$t > s$
Smoothing	(Inferring the past)	$p(h_t\|v_{1:u})$	$t < u$
Likelihood		$p(v_{1:T})$	
Most likely Hidden path	(Viterbi alignment)	$\underset{h_{1:T}}{\operatorname{argmax}}\, p(h_{1:T}\|v_{1:T})$	

Localizing a burglar.





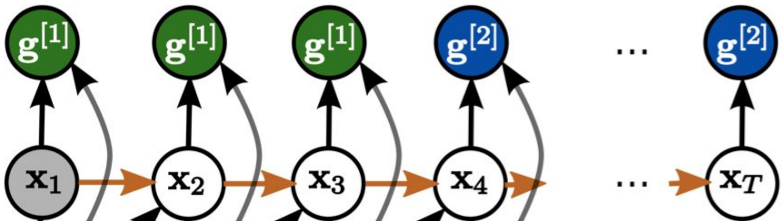
Using Smoothing for robot path planning

$$p(\underline{\boldsymbol{x}}|r=1) = p(\boldsymbol{x}_0)\prod_{t=1}^{T}\mathscr{T}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

Smoothing with neural networks

$$p(\underline{\boldsymbol{x}}|r=1) = \frac{1}{\mathscr{Z}} p(r|\underline{\boldsymbol{x}})\, p(\boldsymbol{x}_0) \prod_{t=1}^{T} \mathscr{T}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

- Cannot be implemented in a Recurrent Neural Network!
- Also the alternative of using 1 Layer per time step is impractical in FF nets.

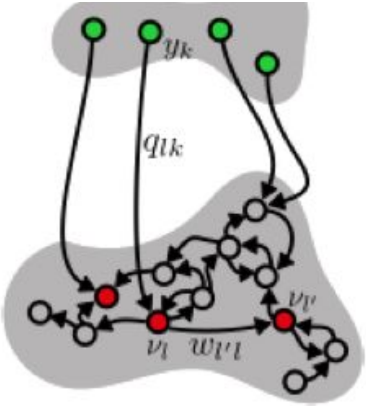


Smoothing in a RNN through forward sampling from a learned distribution

$$p(\underline{\boldsymbol{x}}|r=1) = \frac{1}{\mathcal{Z}} p(r|\underline{\boldsymbol{x}}) \, p(\boldsymbol{x}_0) \prod_{t=1}^{T} \mathcal{T}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

- Reward modulated Hebbian Learning
- Supervised Model Learning (CD)

Neural Planning



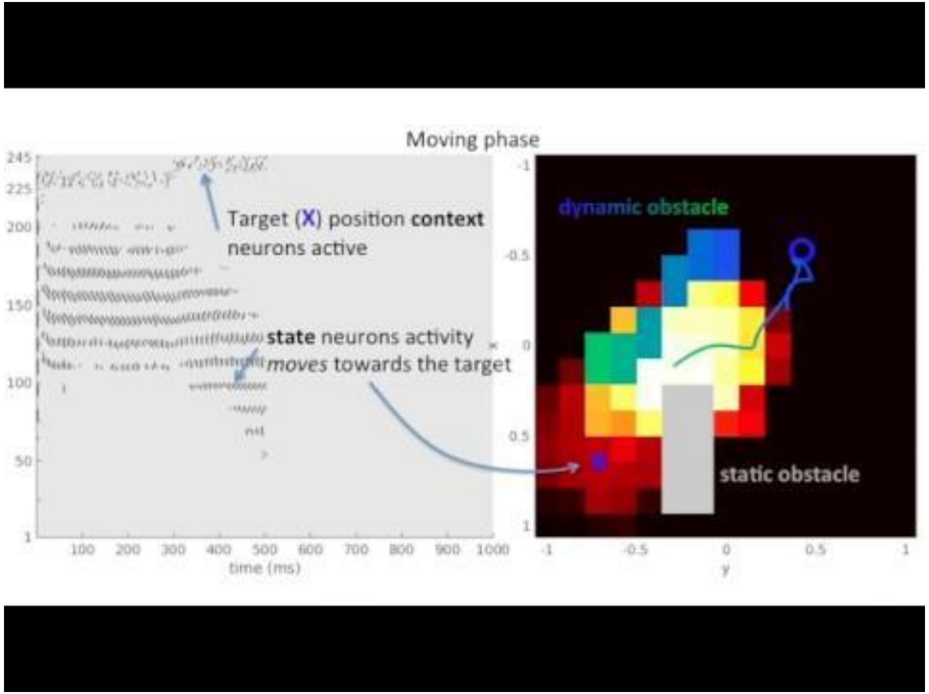
$$q(\underline{\boldsymbol{\nu}}; \boldsymbol{\theta}) = p(\boldsymbol{\nu}_0) \prod_{t=1}^{T} \prod_{k=1}^{K} \rho_{t,k}^{\ \nu_{t,k}} \left(1 - \rho_{t,k}\right)^{1-\nu_{t,k}}$$

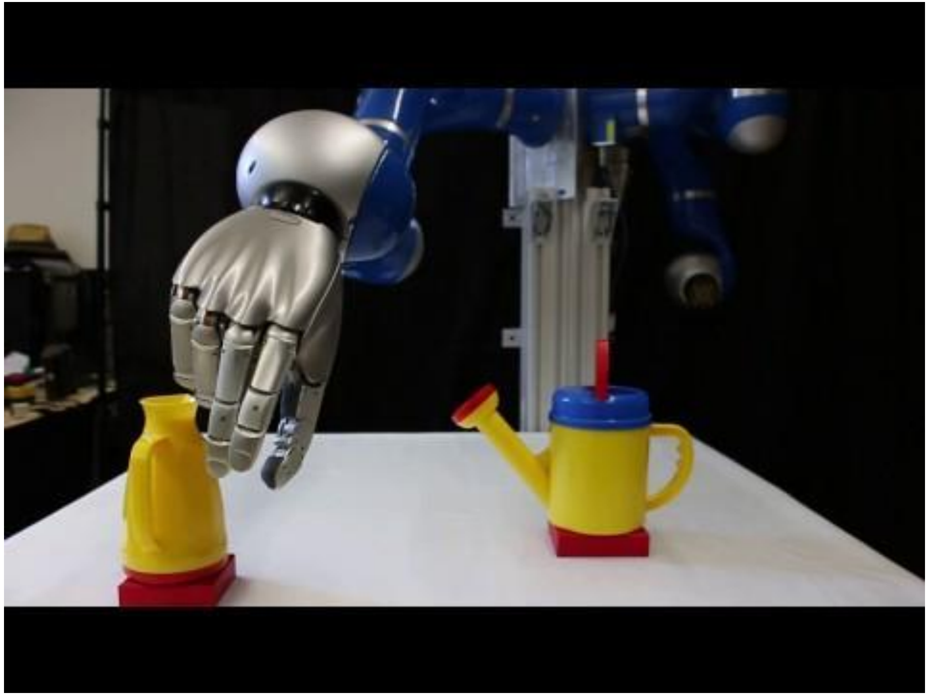
$$= p(\mathbf{v}_0) \prod_{t=1}^{T} \mathcal{T}(\mathbf{v}_t \mid \mathbf{v}_{t-1}) \phi_t(\mathbf{v}_t; \boldsymbol{\theta})$$

$$\mathcal{T}(\mathbf{v}_t \mid \mathbf{v}_{t-1}) = \exp\left(\sum_{i=1}^{K} w_{ki} \, v_{t-1,i} \, v_{t,k}\right)$$

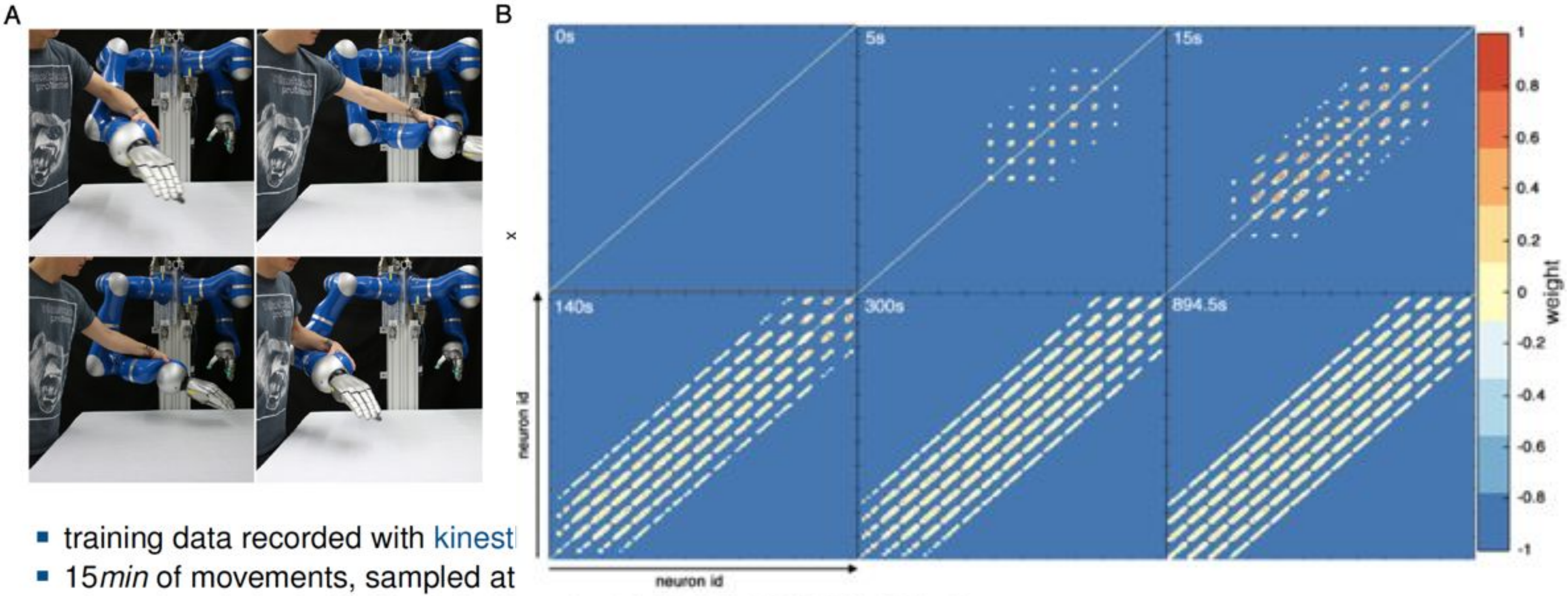
$$\phi_t(\mathbf{v}_t; \boldsymbol{\theta}) = \frac{\exp\left(\sum_{j=1}^{N} \theta_{kj} \, y_{t-1,j} \, v_{t,k}\right)}{\sum_{l=1}^{K} \exp(u_{t,l})}$$

For real robot control without smoothing



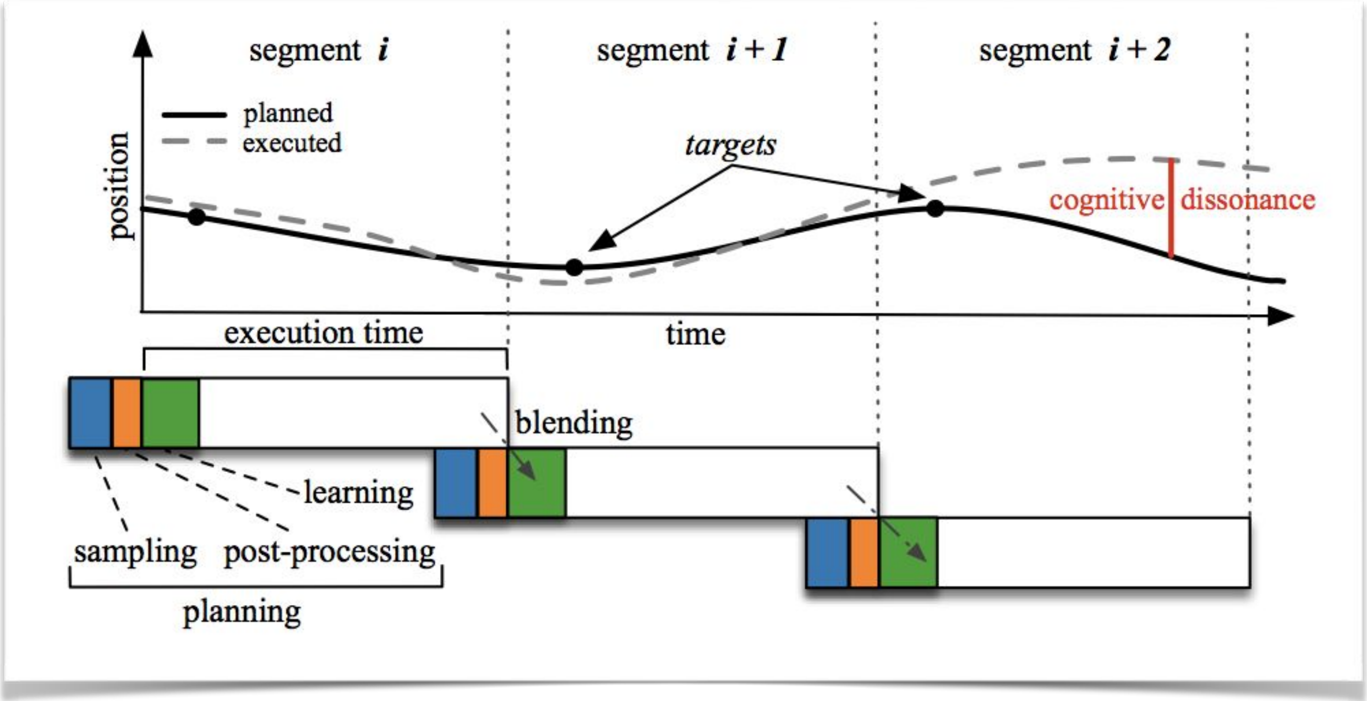


Model Learning in 15 Minutes

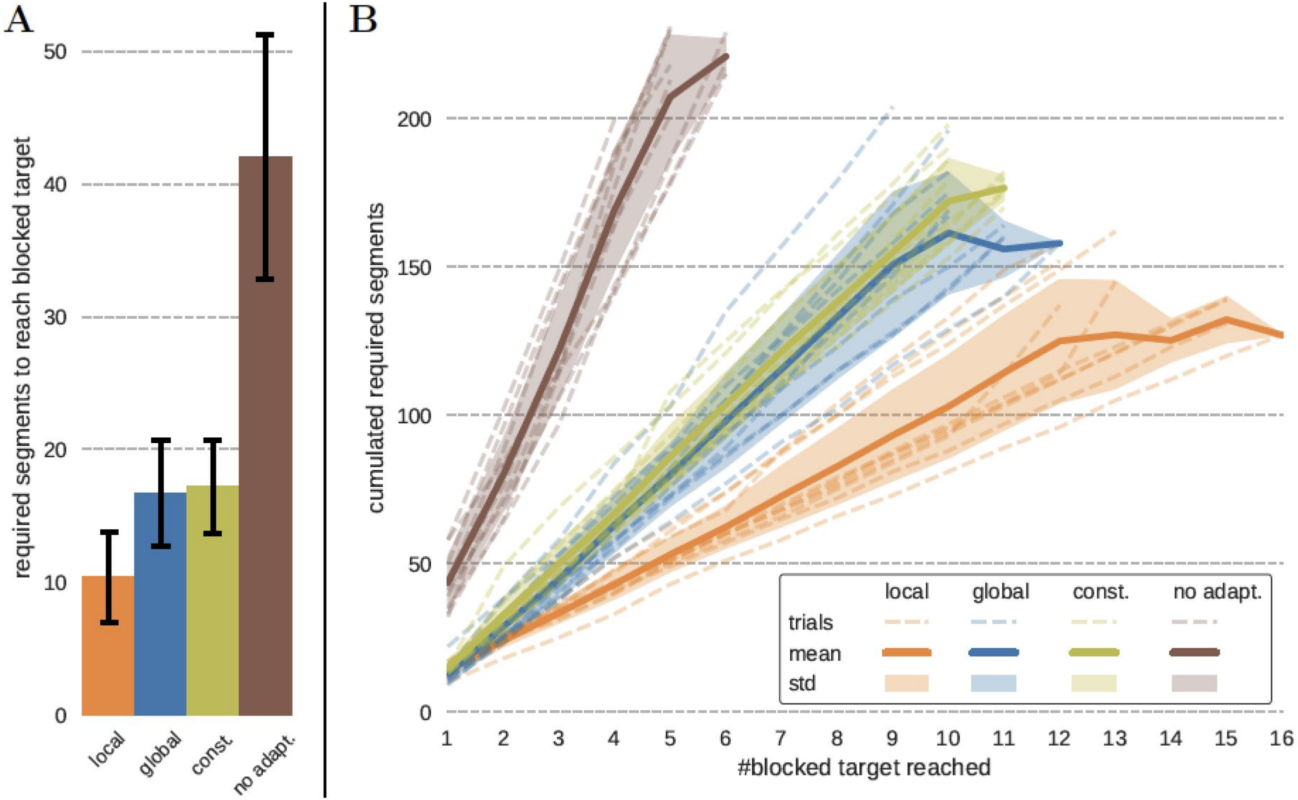


- training data recorded with kinest
- 15*min* of movements, sampled at

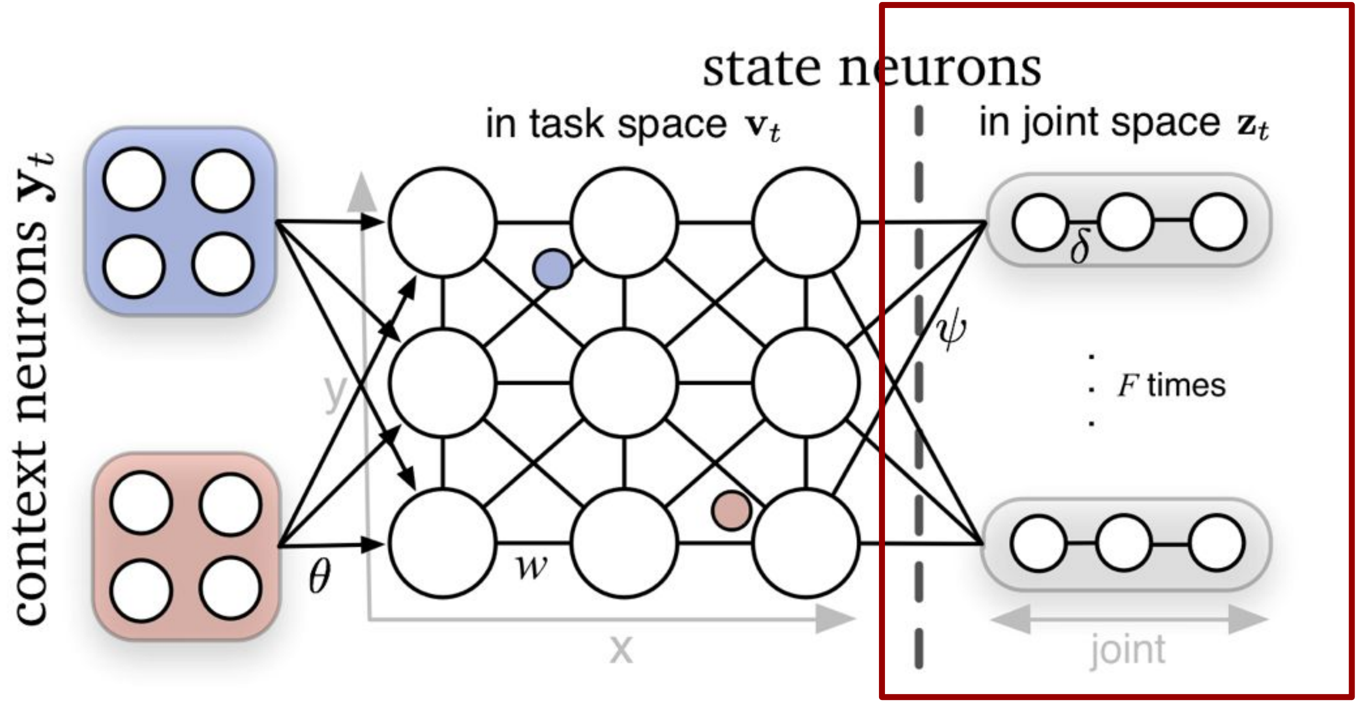
Real Time Adaptation and Control



Efficiency evaluation



Factorized population codes for > 2 dimensions



more at: https://rob.ai-lab.science/publications/

Tanneberg, Daniel; Peters, Jan; Rueckert, Elmar
Intrinsic Motivation and Mental Replay enable Efficient Online Adaptation in Stochastic Recurrent Networks Journal Article
Neural Networks - Elsevier, 2018, (Impact Factor of **7.197** - 2017).

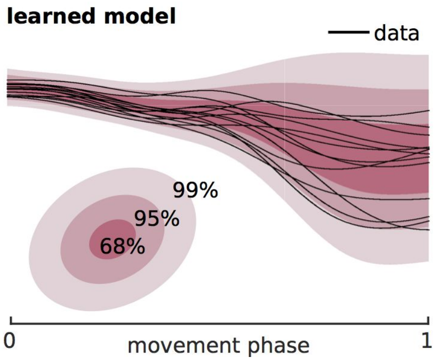
Sosic, Adrian; Rueckert, Elmar; Peters, Jan; Zoubir, Abdelhak M; Koeppl, Heinz
Inverse Reinforcement Learning via Nonparametric Spatio-Temporal Subgoal Modeling Journal Article
Journal of Machine Learning Research (JMLR), 2018.

Rueckert, Elmar; Kappel, David; Tanneberg, Daniel; Pecevski, Dejan; Peters, Jan
Recurrent Spiking Networks Solve Planning Tasks Journal Article
Nature Publishing Group: Scientific Reports, 6 (21142), 2016, (Impact Factor of **4.122** - 2017)

Rueckert, Elmar; Neumann, Gerhard; Toussaint, Marc; Maass, Wolfgang
Learned graphical models for probabilistic planning provide a new class of movement primitives Journal Article
Frontiers in Computational Neuroscience, 6 (97), 2013.

Summary

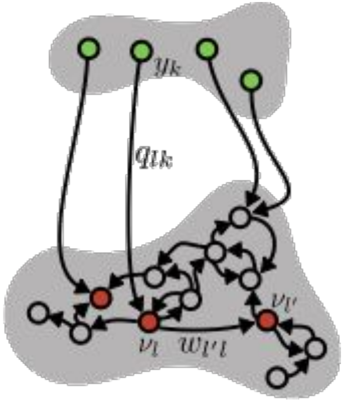
1. How can humans learn new motor skills within few trials?



Learning probabilistic generative models that capture the correlations of multiple joints/signals.

- For **noisy** and **high** dimensional **human** and **robot** data.
- Can exploit **correlations** for **predictions.**
- Low dimensional **feature** representation for **learning.**
- Generative model of **stroke-based** and **rhythmic** movements with **feedback**.

Summary



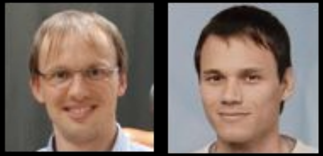
1. How do humans solve cognitive reasoning tasks in huge spaces?

Learning stochastic neural networks grounded in the probabilistic inference framework.

- Simultaneously learning **forward, inverse kinematics** and **state transition models** through kinesthetic teaching.
- Implements **optimal planning** through reinforcement learning.
- **Online adaptation** in few seconds from **intrinsic motivation** signals.
- Model **predictive control** implementation on **real robots**.

- **Darmstadt**: Daniel Tanneberg, Svenja Stark, Gerhard Neumann, Alexandros Paraschos, Roberto Calandra, Jan Peters, Rudolf Lioutikov, Marc Deisenroth, Serena Ivaldi, Tucker Hermans, Philipp Beckerle, Valerio Modugno, Jan Mundo, David Sharma, Jan Kohlschuetter, Svenja Stark, Michael Schmidt, Max Mindt

- **Tübingen**: Moritz Grosse-Wentrup, Martin Giese



- **Ljubijana**: Jan Babic, Jernej Camernik

- **Birmingham**: Michael Mistry, Morteza Azad

- **Graz**: Wolfgang Maass, Robert Legenstein, David Kappel, Dejan Pecevski

Birmingham: Jeremy Wyatt, Michael Mistry, Morteza Azad, **Rome**: Andrea d'Avella and Yuri Ivanenko,
Stuttgart: Marc Toussaint, **Bielefeld**: Thomas Schack, Jochen Steil, **Genua**: Francesco Nori, Lorenzo Natale

More information about the course content..

Books:

- Bishop 2006. **Pattern Recognition and Machine Learning**, *Springer.* — 5 copies at the ZHB
- Barber 2007. **Bayesian Reasoning and Machine Learning**, *Cambridge University Press.* — free online version
- Murray, Li and Sastry 1994. **A mathematical introduction to robotic manipulation**, *CRC Press.* — free online version

Video Lectures:

- videolectures.net *on Gaussian Processes, Inference and Reinforcement Learning*
- coursea.org *on Robotics*

Related lecture notes:

- Humanoid Robotics *by Prof. Dr. Maren Bennewitz, University of Bonn.*
- Lecture notes on learning methods *by Prof. Dr. Marc Toussaint, University Stuttgart.*
- Lecture notes on dynamics *by Prof. Dr. Russ Tedrake, Massachusetts Institute of Technology.*

Thank you for your attention!

Contact:

Universität zu Lübeck
Institute for Robotics and Cognitive Systems
Ratzeburger Allee 160
Building 64, Room 94
23538 Lübeck, Deutschland

Telefon: +49 (0) 451 3101 5209
E-Mail: rueckert@rob.uni-luebeck.de

https://rob.ai-lab.science

Disclaimer: