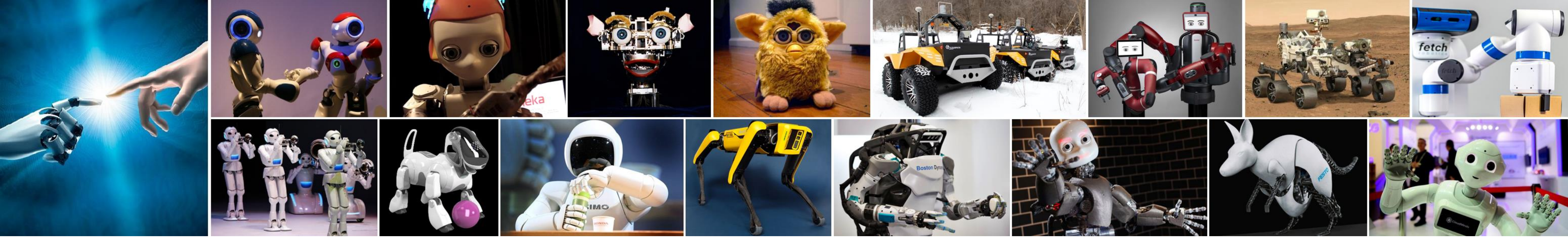


COMPSCI-603: Robotics

Robot Learning

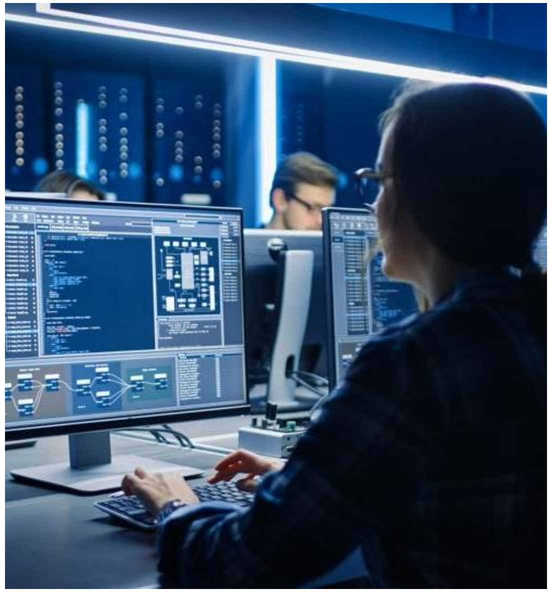
Why Robot Learning?

- Traditional robots were designed for special purposes, e.g., in automotive manufacturing:
  - Welding, assembly, painting / sealing / coating, part transfer, material removal, etc.
- Characteristics:
  - Structured environments.
  - Specific tasks & procedures.
  - Pre-programmed robots.

Why Robot Learning?

General purpose robot

Specific task



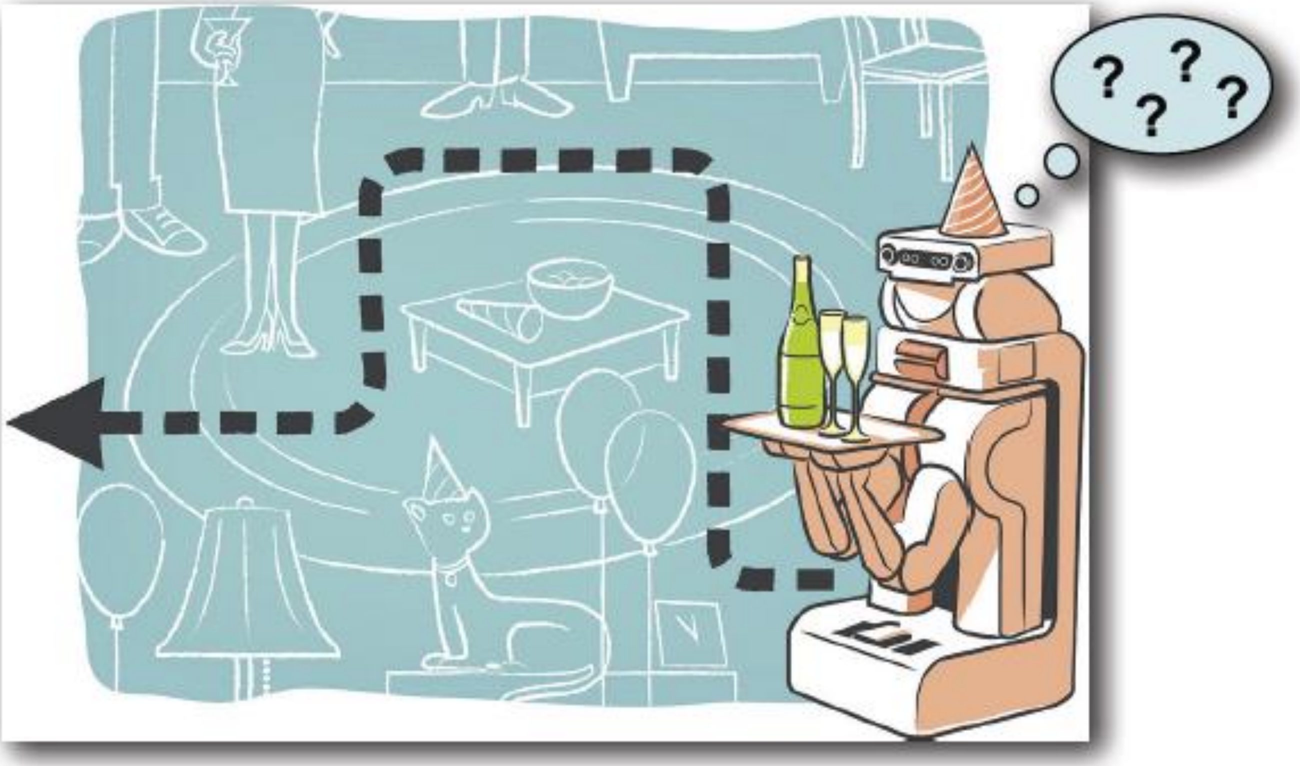
Robotics engineer

Why Robot Learning?

General purpose robot

Why Robot Learning?

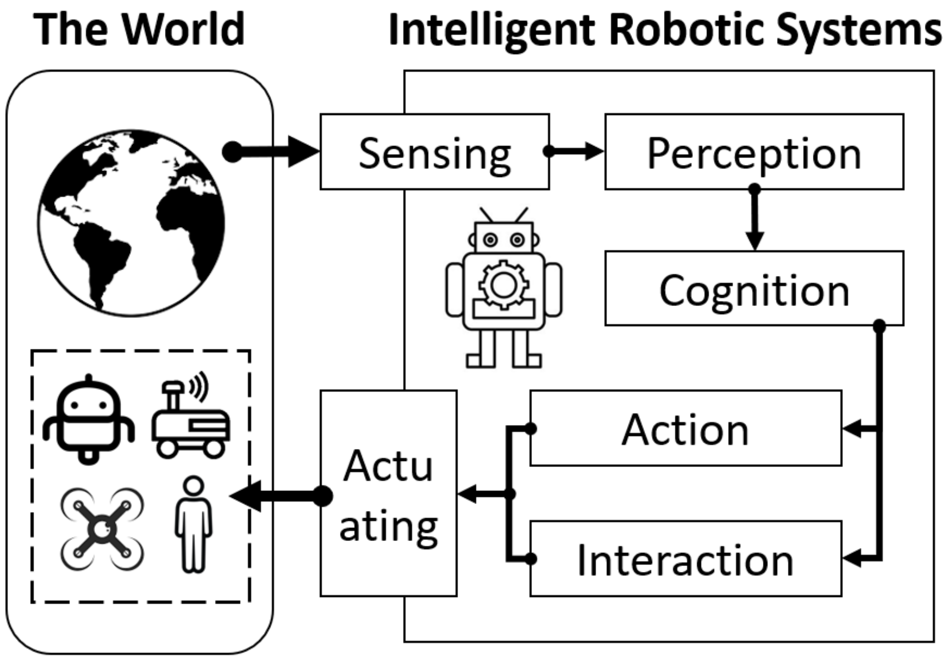
- Programming robots is hard:
  - Huge number of possible tasks that may be changing and new.
  - Tasks difficulty to describe formally.
  - Unstructured environments potentially in open worlds.
  - Humans in the loop.
- Expert engineering may be impractical.



- Robot learning is one of the most promising solutions.

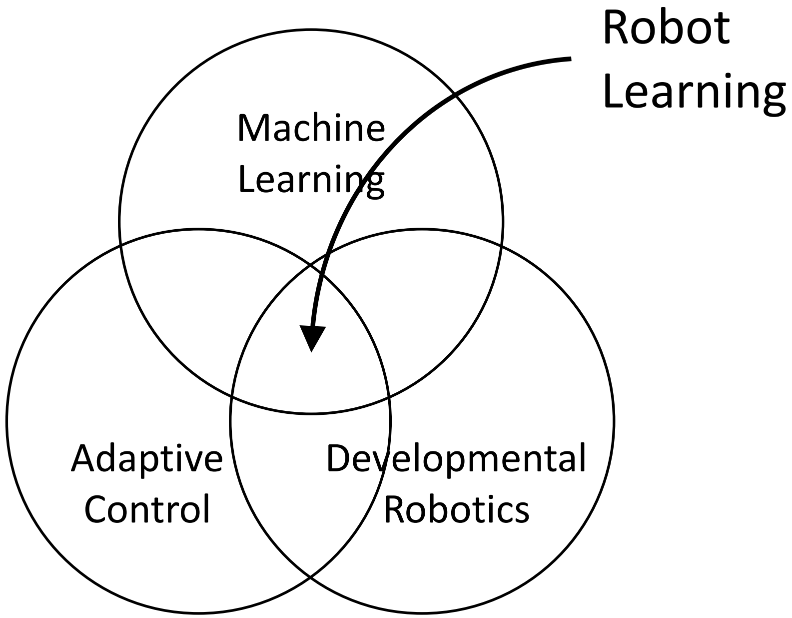
Definition and Scope

- Robot learning is a subfield in robotics:
  - to study techniques allowing a robot to acquire new skills or adapt to its environment,
  - by training computational models from historical experience.
- Robot learning is considered as one aspect of robot cognition.
- Key challenge in robot learning: bridge perception and action, or close the perception-action loop.



Definition and Scope

- Given the popularity of machine learning, it is safe to say that robot learning applies machine learning within the robotics community.
- However, the above statement is not 100% accurate.
- Robot learning is in the intersection of:
  - Machine learning.
  - Adaptive control: automatically adjusts controller parameters to compensate for changing process conditions.
  - Developmental robotics: studies the developmental mechanisms, architectures and constraints that allow lifelong and open-ended learning of new skills and new knowledge in embodied machines.
    - Can a robot learn like a child?



Definition and Scope

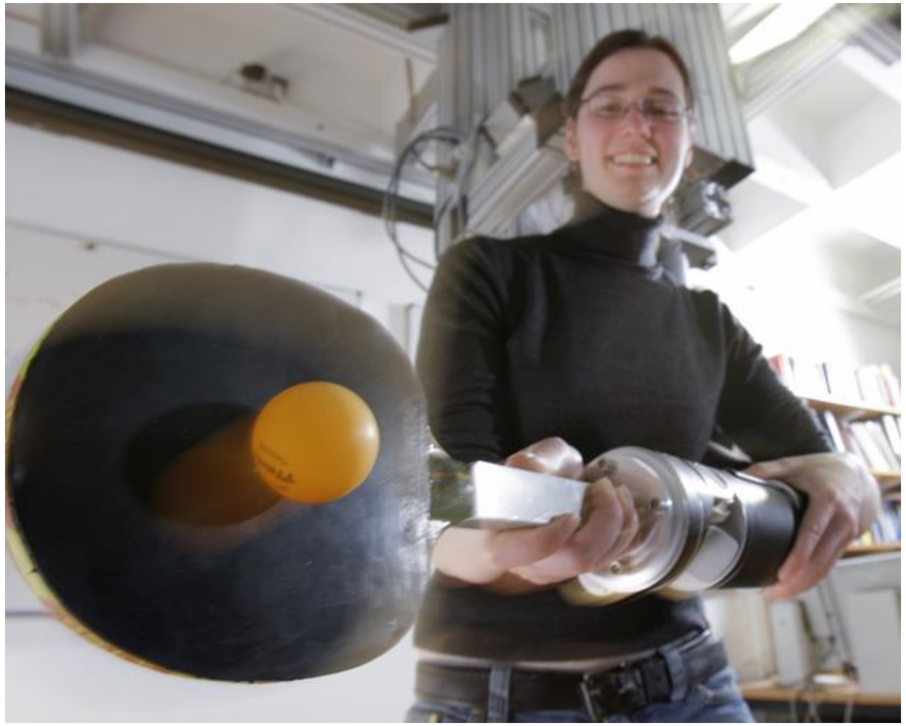
- Broader scope of robot learning (defined by IEEE):
  - learning models of robots, tasks or environments
  - learning deep hierarchies or levels of representations, from sensor and motor representations to task abstractions
  - learning of plans and control policies by imitation and reinforcement learning
  - integrating learning with control architectures
  - methods for probabilistic inference from multi-modal sensory information (e.g., proprioceptive, tactile, vison)
  - structured spatio-temporal representations designed for robot learning such as low-dimensional embedding of movements
  - developmental robotics and evolutionary-based learning approaches

How can a robot learn to perform a task?

Credit: PR2 Robot / Willow Garage

How to teach robot to do a task?

- How to teach a robot to flip pancakes?
- How about teaching a robot to play table tennis with a human?

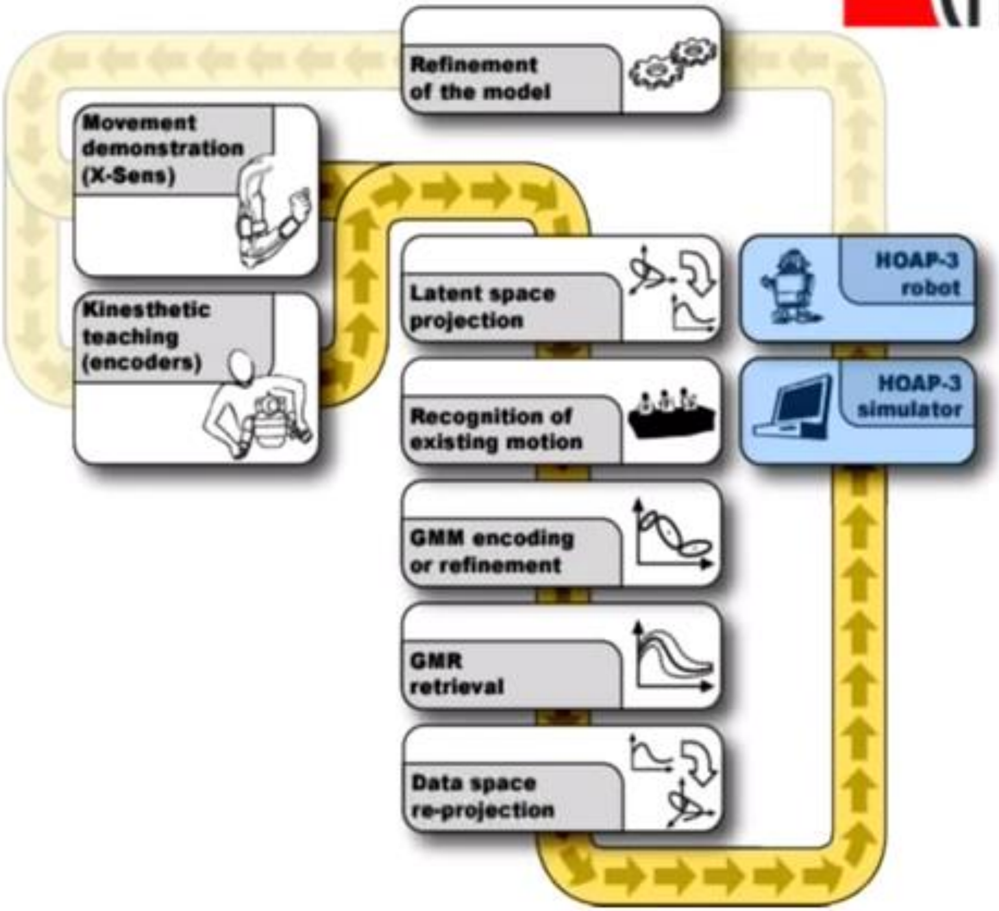


Learning from Demonstration

- Learning from demonstration (LfD) is an end-user development technique for teaching a robot new behaviors by demonstrating the task to transfer directly instead of programming it through machine commands.
- Robot LfD started in the 1980s and has grown steadily in importance.
- At the core, LfD is inspired by the way humans learn from being guided by experts, from infancy through adulthood.
- A large body of work on LfD therefore takes inspiration from concepts in psychology and biology.
- Nowadays, the vast majority of work on LfD follows a machine learning approach.

Learning from Demonstration

- LfD is also called:
  - Programming by Demonstration
  - Imitation Learning
  - Apprenticeship Learning



Reference: Sylvain Calinon and Aude Billard. "Incremental learning of gestures by imitation in a humanoid robot." In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI). 2007.

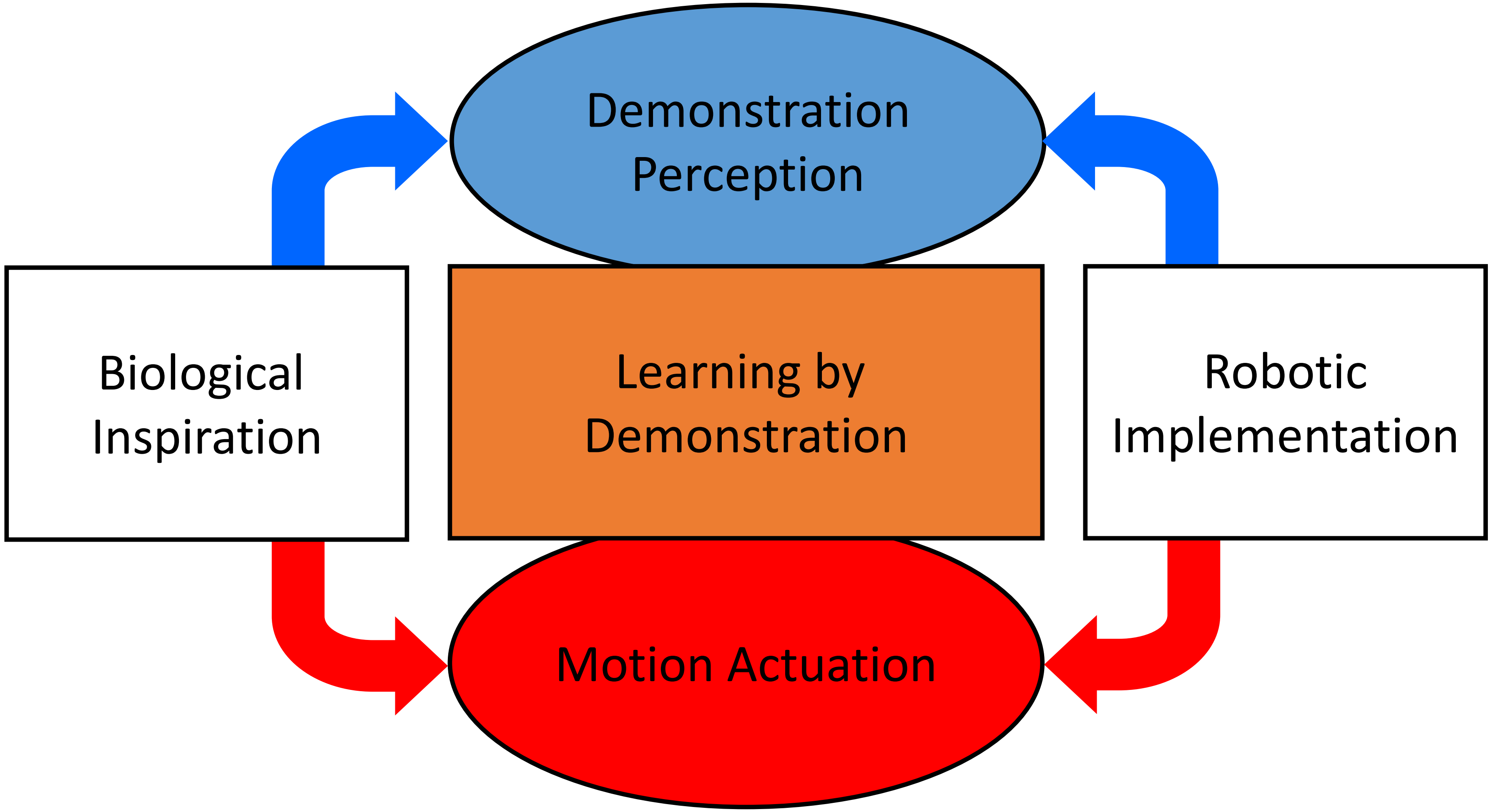
Learning from Demonstration

Demonstration Perception
How are actions perceived?
How is information represented?

Learning by Demonstration
What should be imitated?
Intention, dynamics, or poses?

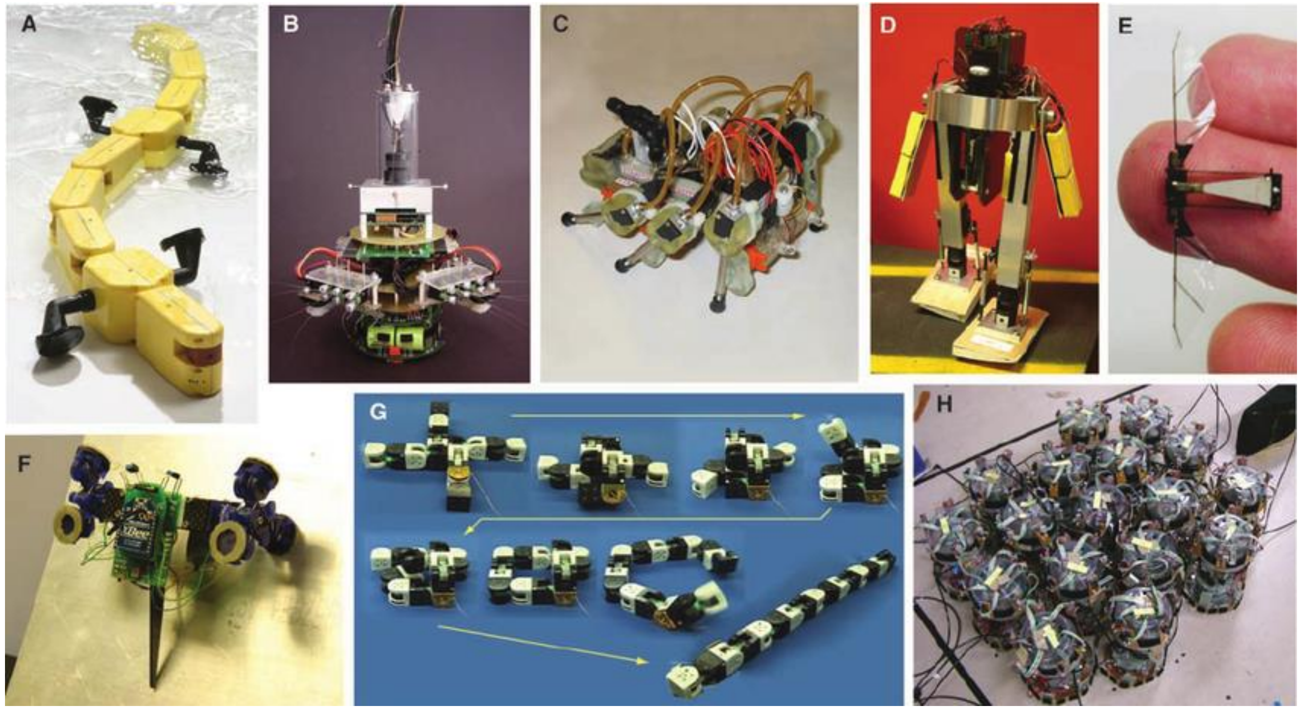
Motion Actuation
How is information transferred and implemented on a physical robot?

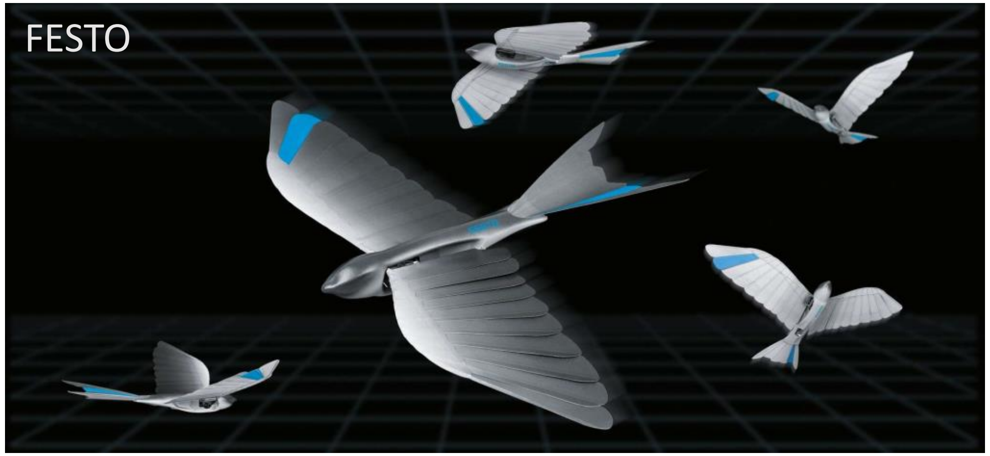
Learning from Demonstration



Learning from Demonstration

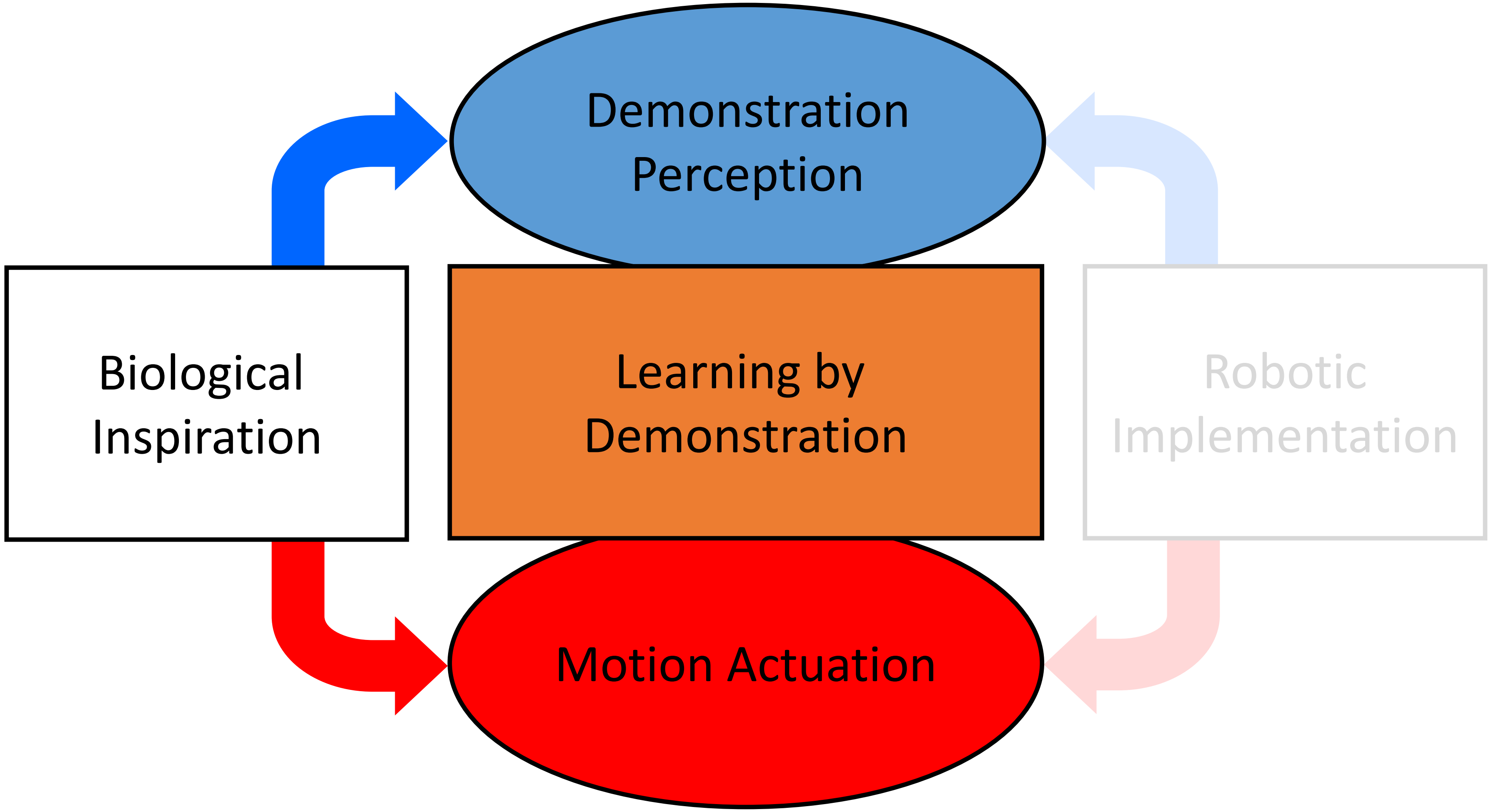
- Prior to building capability in robots, we often want to understand how the equivalent capability works in humans and animals – biological inspiration:





Rolf Pfeifer, Max Lungarella, and Fumiya Iida. "Self-organization, embodiment, and biologically inspired robotics." Science, no. 5853, pp. 1088-1093. 2007.

Learning from Demonstration



Biological Inspiration: Human Imitation

- “True” imitation: Ability to learn new actions not part of the usual repertoire, by humans only, and possibly great apes.

- “True” imitation is differentiated from copying (flocking, schooling, following), stimulus enhancement, or contagion.

Reference: Whiten & Ham, Advances in the Study of Behaviour, 1992.

Biological Inspiration: Human Imitation

- Newborns to 3-month infants: Innate facial imitation.
- Imitating tongue and lips protrusion, mouth-opening, head movements, cheek and brow motion, eye blinking.
- Delayed imitation up to 24 hours.
- Imitation is mediated by a stored representation.

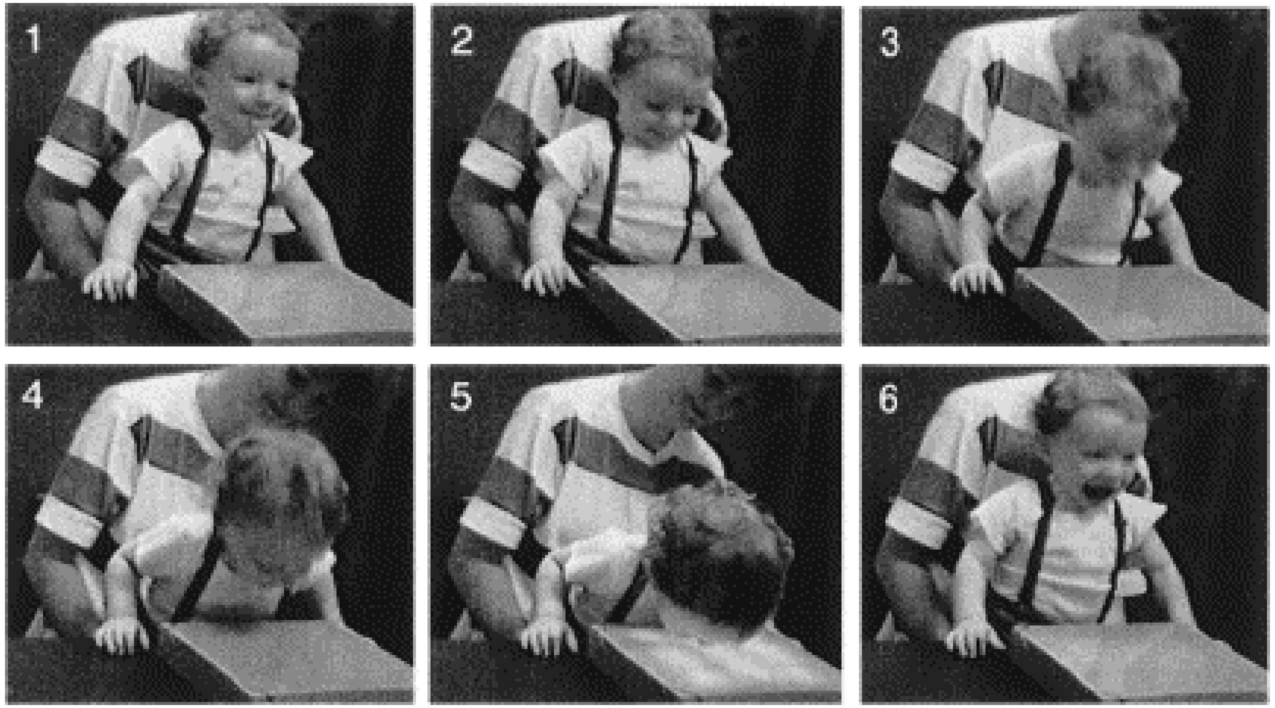
References:
Meltzoff & Moore, Early Development and Parenting, 1997.
Meltzoff & Moore, Developmental Psychology, 1989.

Slide modified from Prof. Aude Billard

Biological Inspiration: Human Imitation

- 9-12-month infants: Deferred and delayed imitation of novel behavior.

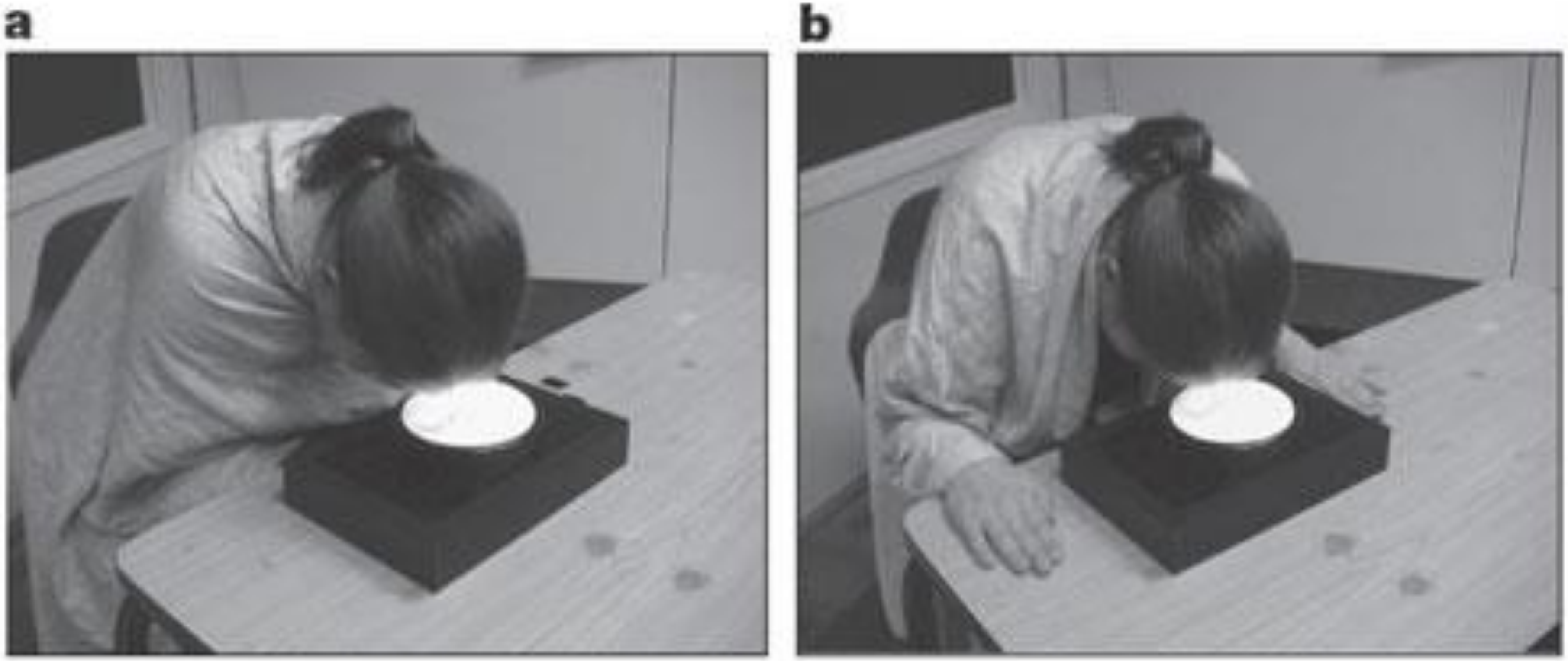
- 67% of the infants who saw the display reproduced the act after the week's delay, as compared to 0% of the control infants who had not seen the novel display.



Slide modified from Prof. Aude Billard

Biological Inspiration: Human Imitation

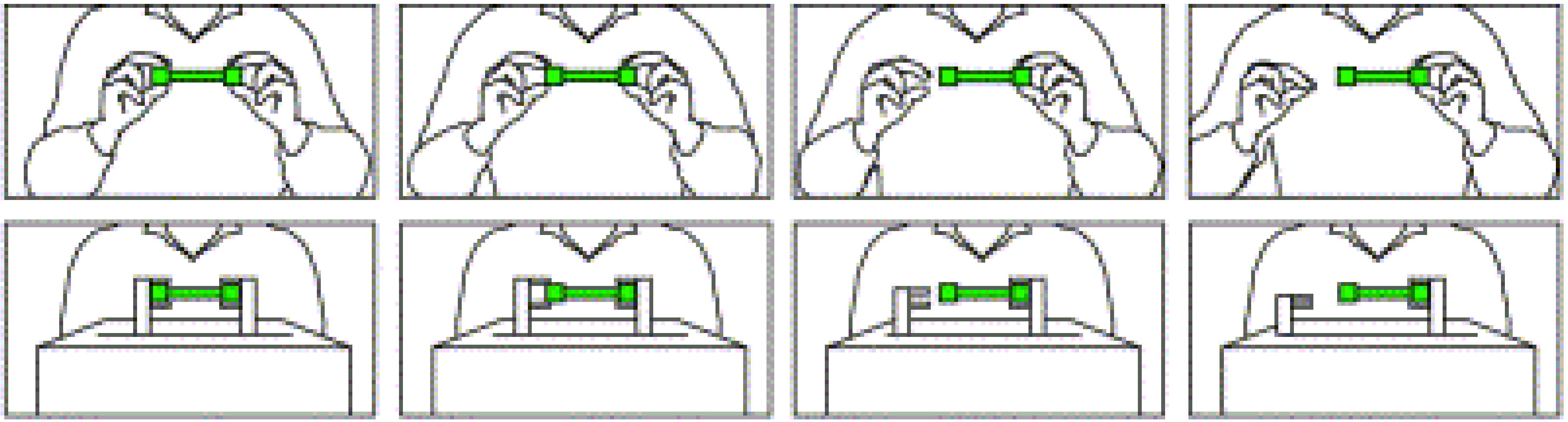
- 14-month infants: Imitation of new action to achieve the same goal only if they consider it to be the most rational alternative.



Slide modified from Prof. Aude Billard

Biological Inspiration: Human Imitation

- 18-month infants:
  - Differentiate between human and machine demonstration.
  - Learn from unsuccessful examples.



Reference: Meltzoff, Dev. Psychol. 31, 1995.

Slide modified from Prof. Aude Billard

Biological Inspiration: Human Imitation

- Children:
  - Imitation is hierarchical and goal-directed.
  - Single-hand motions: accurate ipsilateral imitation, 48% substitution for cross-lateral imitation.
  - Two-hand motions: only 10% substitution for cross-lateral imitation.

  - Two-phase motion eliminates mistakes.
  - Adding constraints of hand gestures increases mistakes.

Biological Inspiration: Human Imitation

- Adults:
  - Imitation reaches the highest level of complexity.
  - Imitation is present in all learning activities.
  - Imitation in adulthood is influenced by movement observation, handedness, orientation of the demonstrator.
  - Social influence in establishing group norms, collective frame of reference, transmission of phobias.

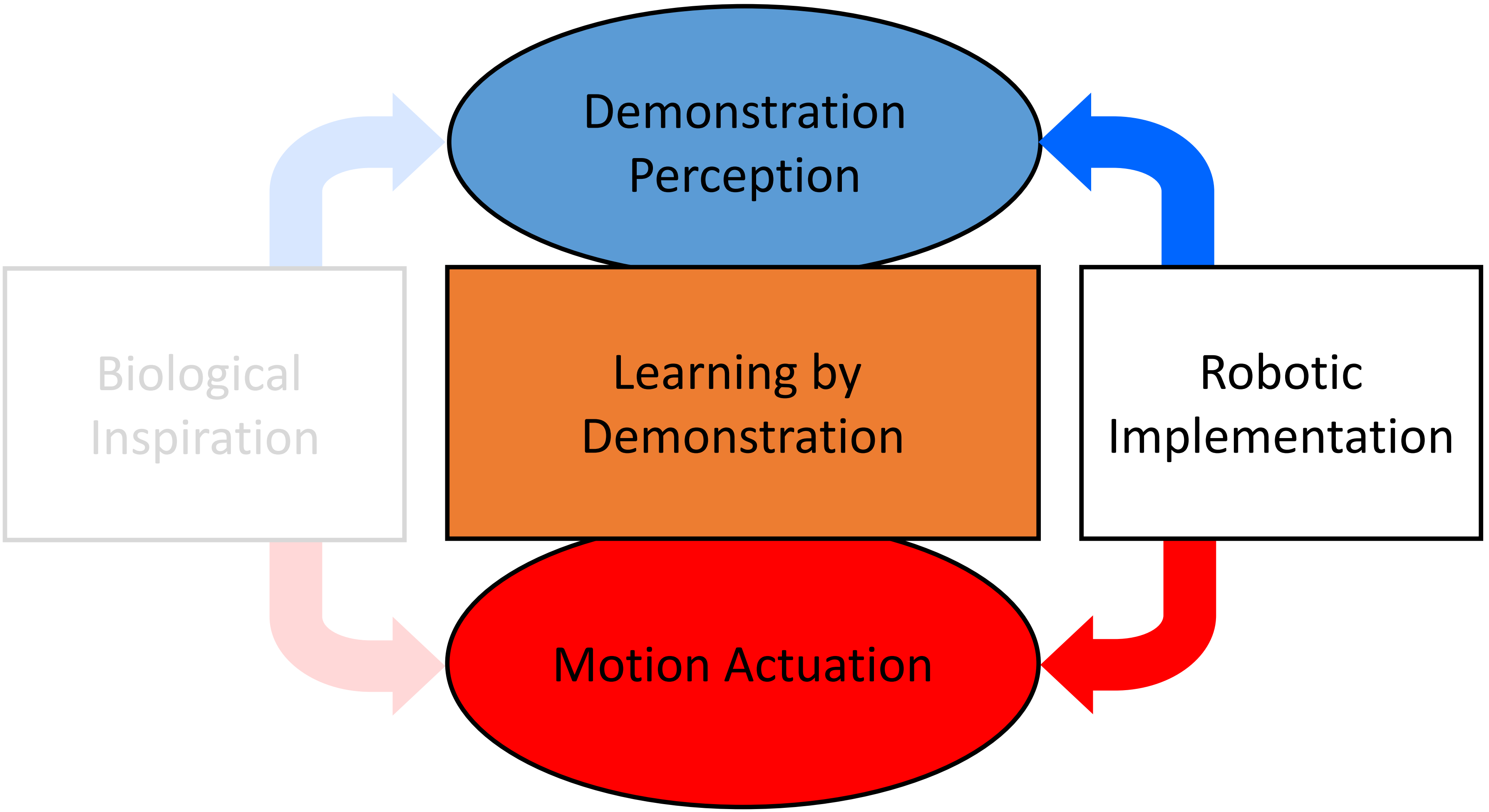
Biological Inspiration: Human Imitation

- Advantages: When is imitation useful?
  - It is a powerful paradigm of transferring skills to perform tasks.
  - It speeds up the learning process by showing possible solutions or conversely by showing bad solutions.

- Disadvantages: When is imitation not useful?
  - Inappropriate: When a good solution for the teacher is not a possible solution for the learner (when not considering adaptation and reinforcement).
  - Disadvantageous: When it induces you in error from a bad teacher.

FESTO Robotic Bird: https://www.youtube.com/watch?v=Fg_JcKSHUtQ

Learning from Demonstration

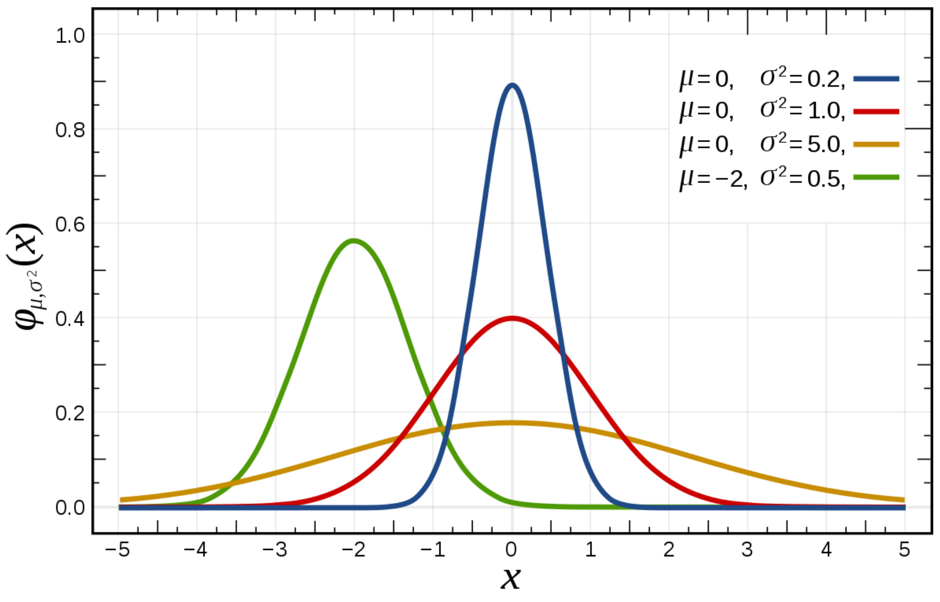


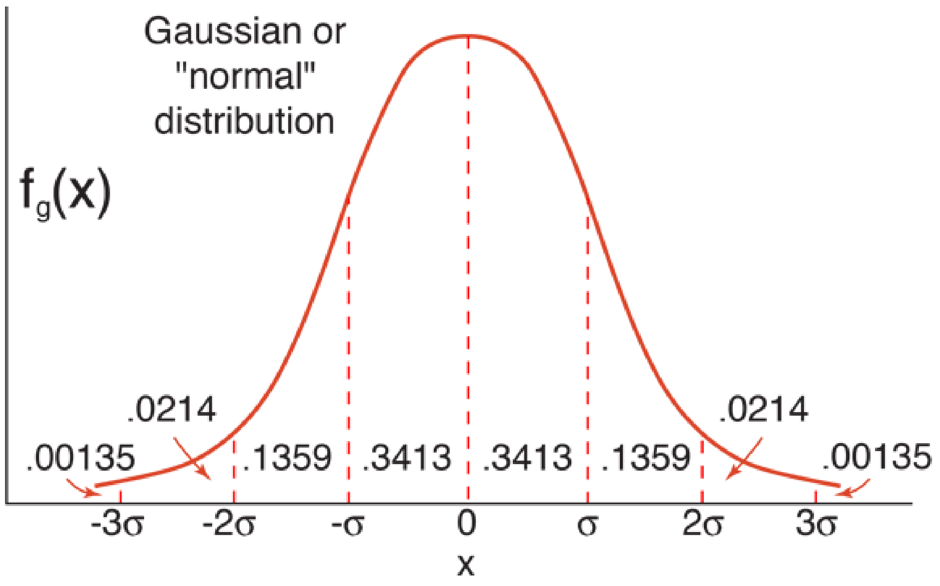
Reference: Sylvain Calinon and Aude Billard. "Incremental learning of gestures by imitation in a humanoid robot." In Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI). 2007.

Mathematical Background

- 1-D Gaussian (normal) distribution has a characteristic symmetric bell curve that quickly falls off towards 0.

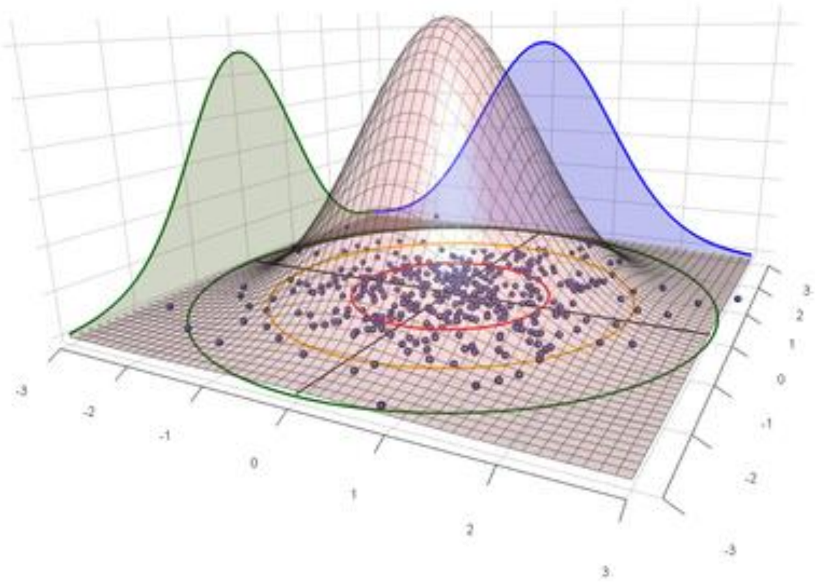
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$





Mathematical Background

- Multivariate Gaussian distributions in the n-D space $\mathcal{N}(x|\mu,\Sigma)$:



$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

$$\mu = \mathbb{E}(X) \qquad \Sigma = \mathtt{Cov}(X) = \mathbb{E}[(X-\mu)(X-\mu)^T]$$

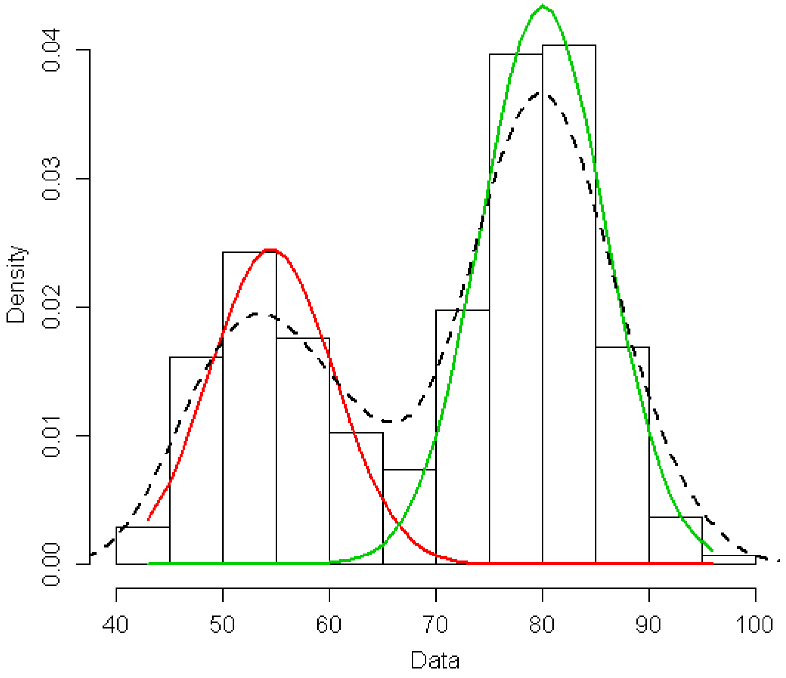
- Given a multivariate Gaussian distribution, its marginals, conditionals, and linear transformations are also Gaussian.

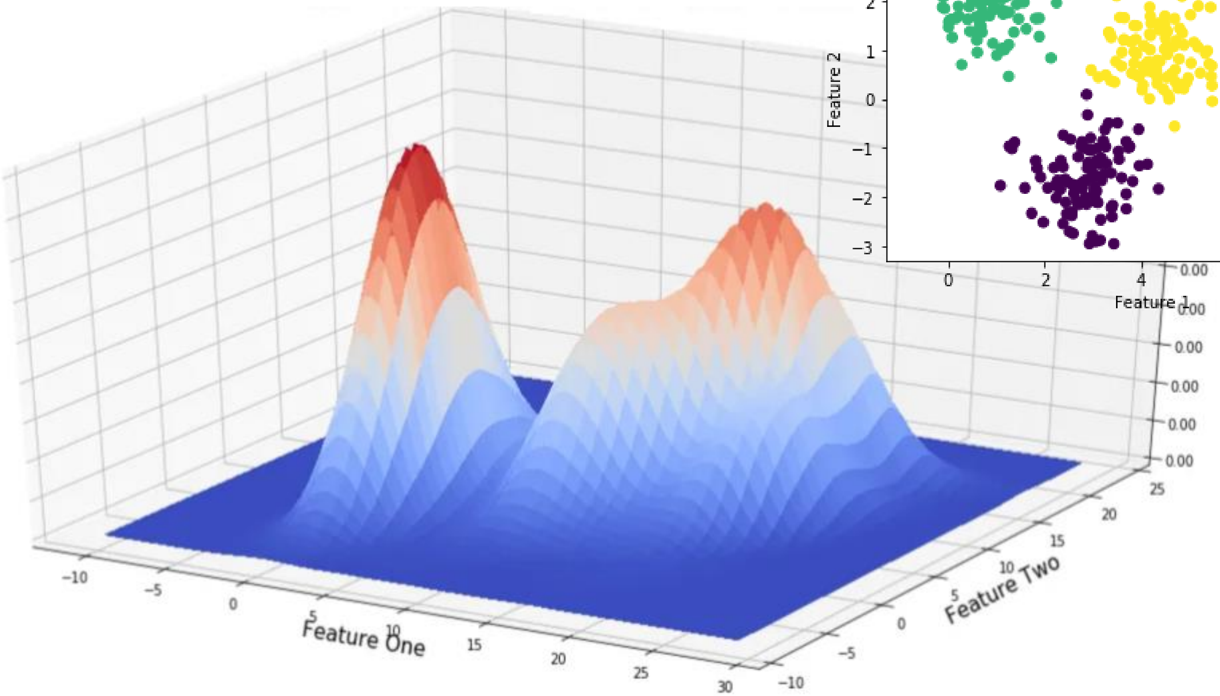
Mathematical Background

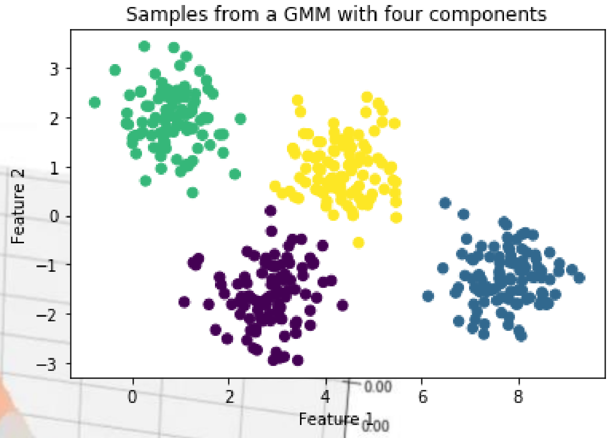
- Gaussians are very common in probability theory and important in statistics, which are also widely used in machine learning.
- Physical quantities that are expected to be the sum of many independent processes often have distributions that are nearly Gaussian (e.g., sensor noise).
- Gaussians are useful because of the **central limit theorem**:
    - Taking sufficiently large independent and identically distributed (i.i.d.) samples, the distribution of the samples will be approximately normally distributed.

Mathematical Background

- Gaussian Mixture Models (GMM)
  - A mixture model is a probabilistic model, which assumes the underlying data belongs to a mixture distribution.







Mathematical Background

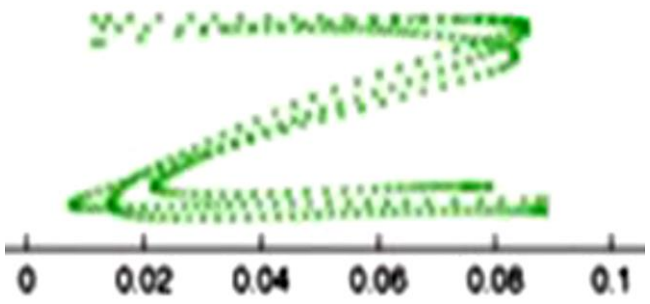
- Gaussian Mixture Models (GMM)
  - GMM computes the probability using a mixture of K Gaussians:

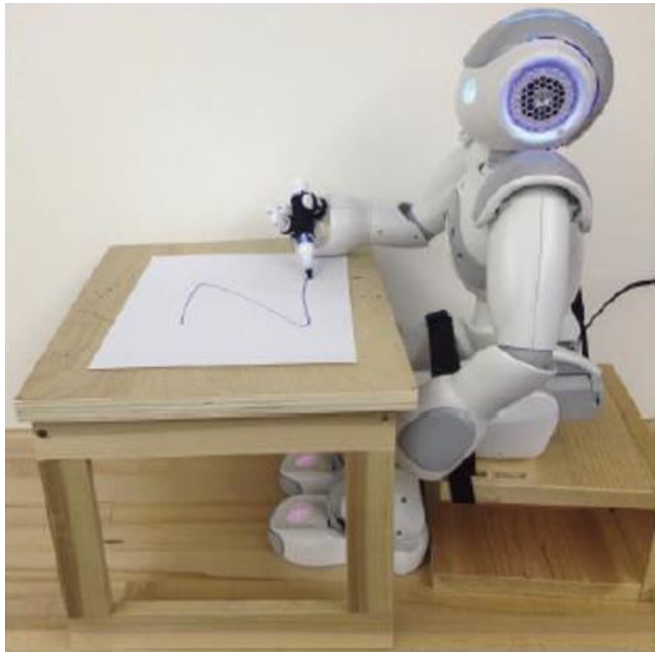
$$p(\boldsymbol{x}) = \sum_{i=1}^{K} w_i \cdot \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \text{where} \quad \sum_{i=1}^{K} w_i = 1, \quad 0 \leq w_i \leq 1$$

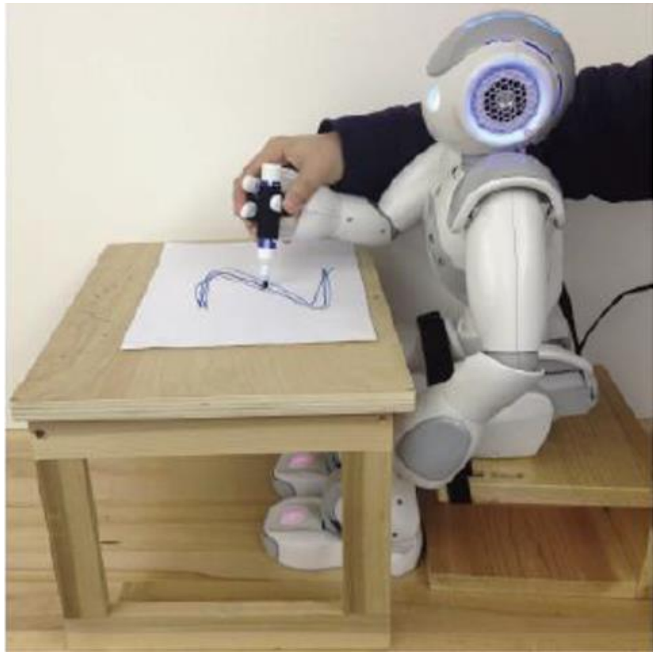
  - GMM can generate data points (samples) in two steps:
    - Select which component i the data point belongs to according to the multinomial distribution of $(w_1, \dots, w_K)$.
    - Generate the data point according to the probability of the i-th component.
- Gaussian Mixture Regression (GMR)
  - Given a GMM, a GMR is used to compute the conditional distribution to generate data that satisfies certain condition, e.g., $p(\boldsymbol{x}_j|\boldsymbol{x}_{-j})$.

LfD by GMM and GMR

- In this example, kinesthetic demonstrations (e.g., a sequence of locations), $\{t, s\}$, are provided by holding the robot's arm to draw.







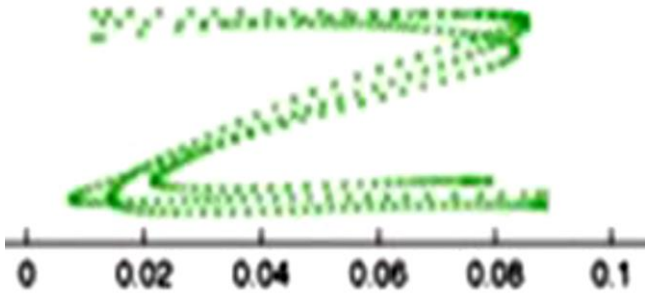
LfD by GMM and GMR

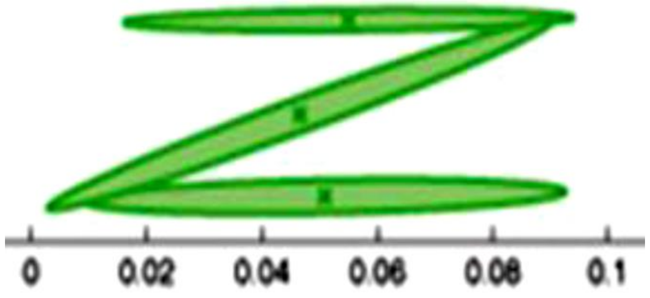
- Demonstrations can be modeled as GMMs.

$$p(t,s)=\sum_{k=1}^{K} w_k \mathcal{N}_k\left(t,s|\mu_{ts\,k},\Sigma_{ts\,k}\right)$$

$$\mu_{ts}=\begin{pmatrix}\mu_t\\ \mu_s\end{pmatrix}$$

$$\Sigma_{ts}=\begin{pmatrix}\Sigma_{tt} & \Sigma_{ts}\\ \Sigma_{st} & \Sigma_{ss}\end{pmatrix}$$





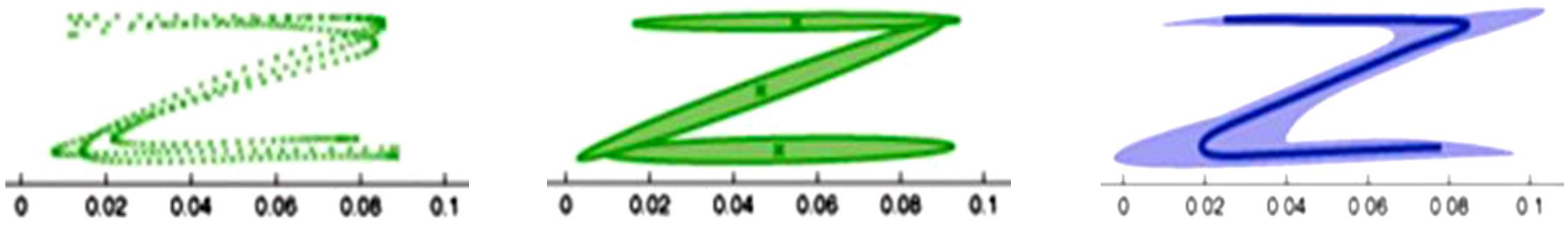
$$\mu_{s|t}=\mu_s+\Sigma_{st}\Sigma_{tt}^{-1}(t-\mu_t)$$

$$\Sigma_{s|t}=\Sigma_{ss}-\Sigma_{st}\Sigma_{tt}^{-1}\Sigma_{ts}$$

LfD by GMM and GMR

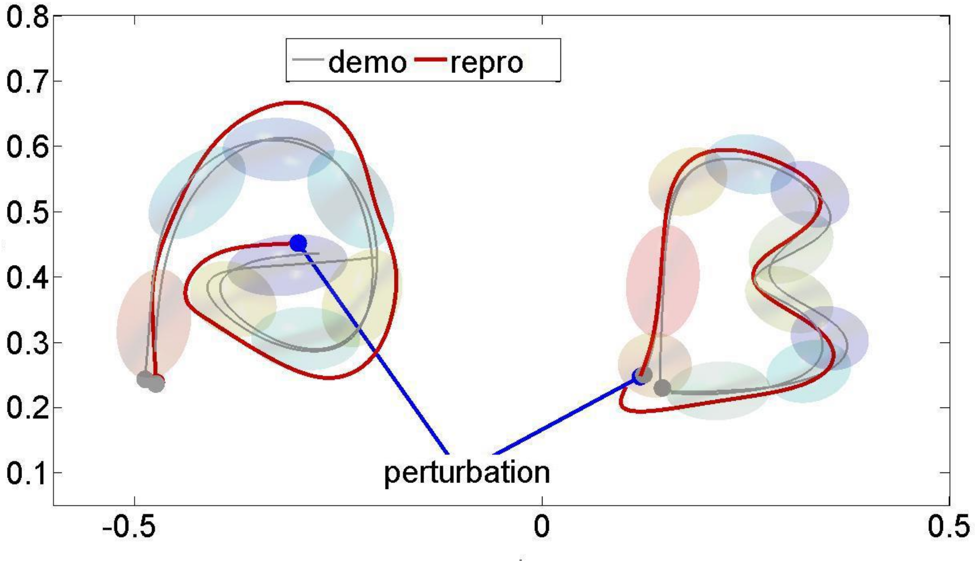
- GMR is used to retrieve the trajectory, namely the expected position at each time step:

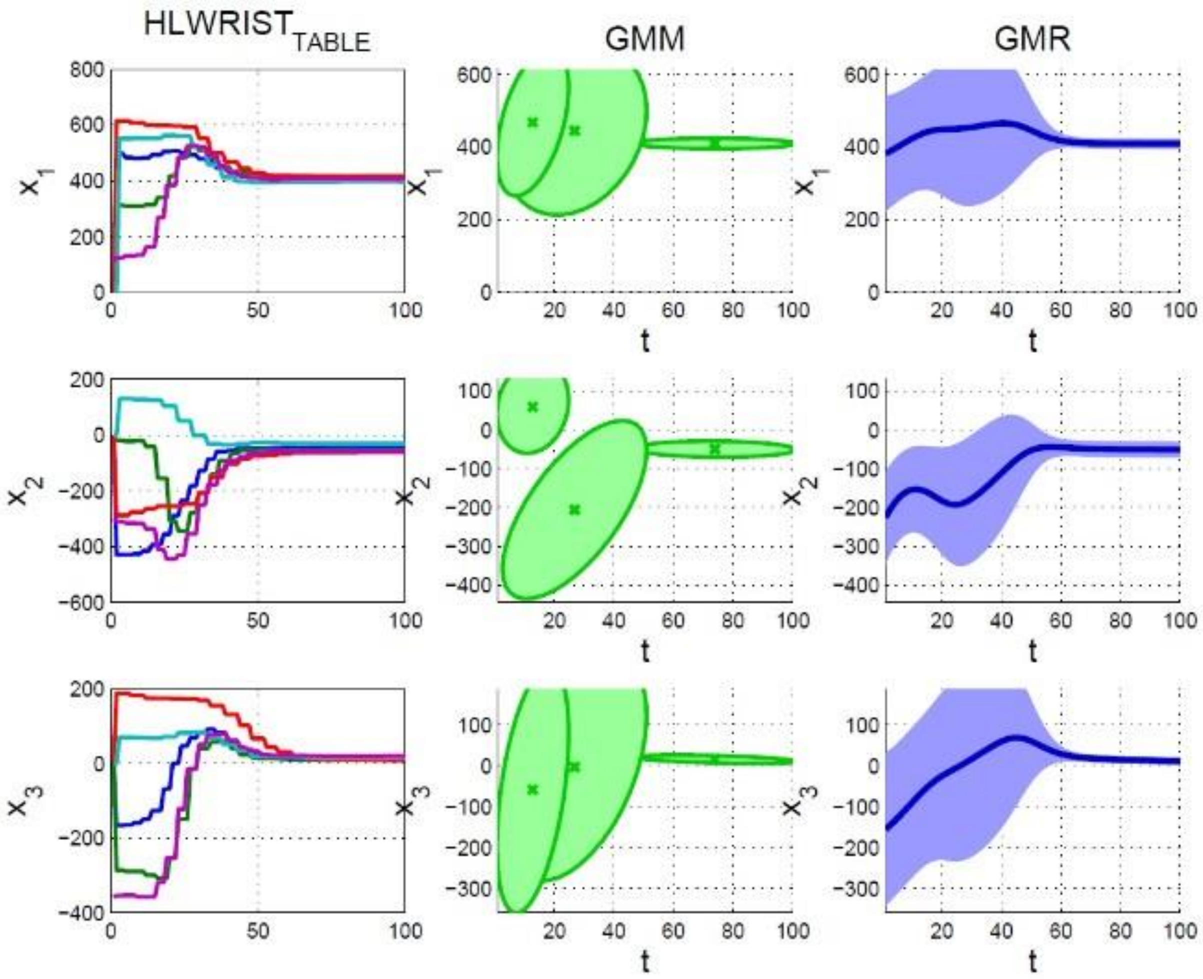
$$p(s|t) = \sum_{k=1}^{K} w_{s|t\,k} \mathcal{N}_k(s|\mu_{s|t\,k}, \Sigma_{s|t\,k}) \qquad w_{s|t\,k} = \frac{\mathcal{N}_k(t|\mu_{t\,k}, \Sigma_{t\,k})}{\sum_{j=1}^{K} \mathcal{N}_j(t|\mu_{t\,j}, \Sigma_{t\,j})}$$



LfD by GMM and GMR

- Examples of grasping:
  - GMM encodes the trajectory.
  - GMR retrieves the trajectory.
- Robustness to perturbations:





LfD and RL for Learning Robot Table Tennis: https://www.youtube.com/watch?v=SH3bADiB7uQ

Problems to Implement GMM/GMR-based LfD

- How to provide demonstrations to a robot?

- How to estimate the parameters of a Gaussian or GMM?
  - Using data for learning or training computational models
- How to estimate the number of Gaussian component in a GMM?
  - Deciding hyperparameter values
- How to align the demonstrated trajectories with different speed?
  - Data preprocessing

Providing Demonstrations

(a) Kinesthetic (b) Teleoperation (c) Observation

Demonstration	Ease of Demonstration	High DOF	Ease of Mapping
Kinesthetic	✓		✓
Teleoperation		✓	✓
Observation	✓	✓	

Ravichandar, Harish, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. "Recent advances in robot learning from demonstration." Annual Review of Control, Robotics, and Autonomous Systems, issue.3, pp.297-330, 2020.

Estimating Parameters

- To estimate parameters of a Gaussian, we may use maximum-likelihood estimation (MLE) to find the parameters under which the data is most likely for that model:
  - Likelihood function:

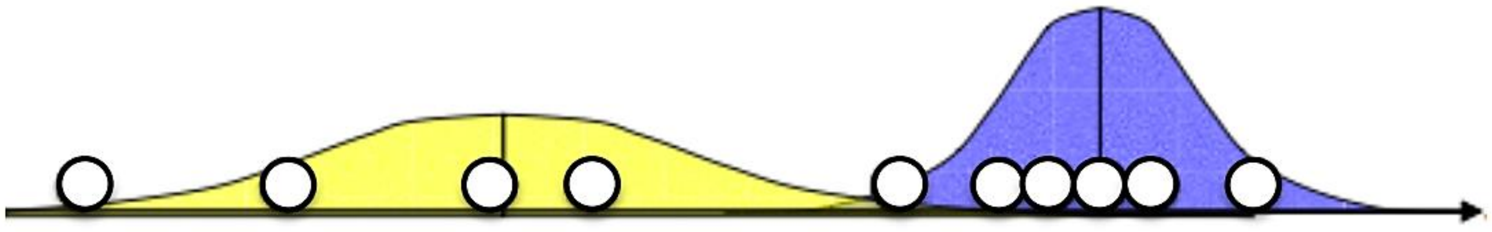
$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X})$$

  - The likelihood is thought of as a function of the parameters Θ where the data $\mathcal{X}$ is fixed.
  - In the MLE problem, our goal is to find the Θ that maximizes $\mathcal{L}$ or log of $\mathcal{L}$:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \; \mathcal{L}(\Theta|\mathcal{X})$$

Estimating Parameters

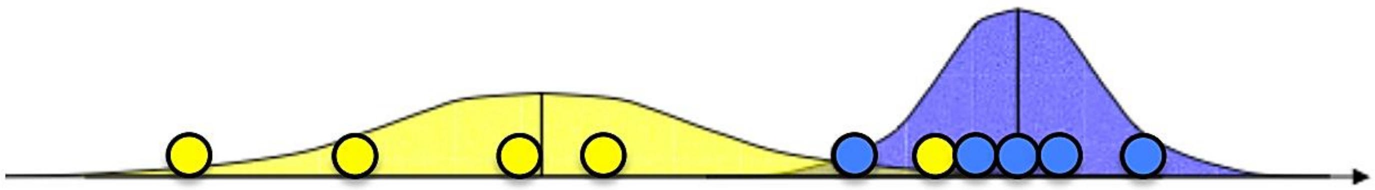
- Does MLE work for GMMs?
  - The answer is no…
  - Since the data points are not from the identical Gaussian components.
- To estimate parameters with hidden variables, we may use the classic Expectation-maximization (EM) algorithm:
  - EM is an iterative method to find maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables.



Credit: Victor Lavrenko

Estimating Parameters

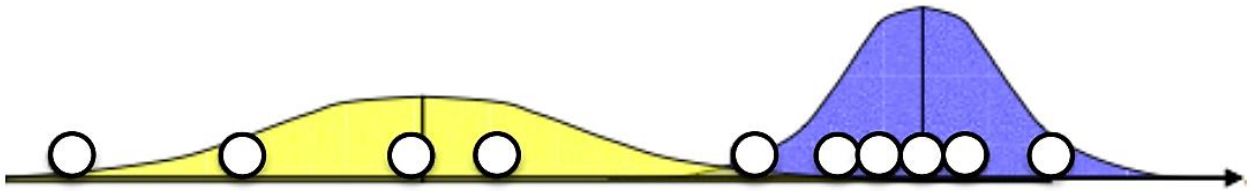
- Given measurements $x_1, \ldots, x_n$:
  - $K = 2$ components with unknown parameters.
  - If the source of each observation is known, estimation is trivial.



$$\mu_b = \frac{x_1 + x_2 + \ldots + x_{n_b}}{n_b}$$

$$\sigma_b^2 = \frac{(x_1 - \mu_1)^2 + \ldots + (x_n - \mu_n)^2}{n_b}$$

  - If we know parameters of the Gaussians, we can estimate which component that each observation comes from.



$$P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2} \right)$$

Credit: Victor Lavrenko

Estimating Parameters

- With hidden variables, it is a chicken and egg problem:
  - We need (μ_a, σ_a^2) and (μ_b, σ_b^2) to estimate the source of the observations.
  - We need to know the source to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2).
- EM algorithm overview:
  - Start with randomly initialization of the Gaussians (μ_a, σ_a^2) and (μ_b, σ_b^2).
  - **E-step:** for each observation x_i, compute $P(a|x_i)$ and $P(b|x_i)$ to estimate which Gaussian component it comes from.
  - **M-step:** update (μ_a, σ_a^2) and (μ_b, σ_b^2) of the Gaussians to fit points assigned to them.
  - Iterate until convergence.

Estimating Hyperparameters

- Broadly, estimating # Gaussian components in GMMs is a hyperparameter estimation or model selection problem.
- Model Selection: Given different models defined by different hyper-parameter values, select the best model (i.e., the hyperparameter resulting in best performance).
- Many methods exist based on different criteria:
  - Cross-validation methods: use different portions of the data to train and validate a model.
  - Information-based methods, e.g., Bayesian information criterion (BIC): balance between likelihood and model complexity.
- Occam's razor: pick "simplest" of all models that fit.

Aligning Trajectories

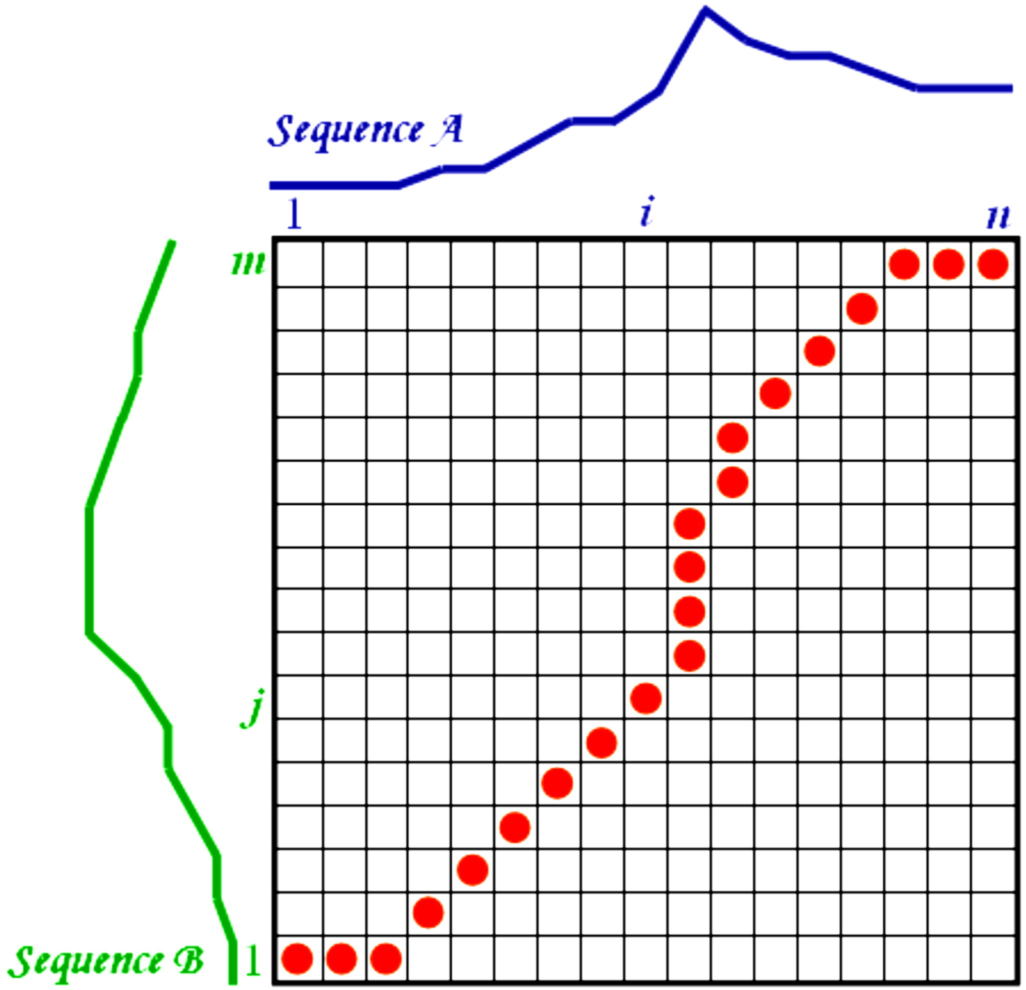
- Trajectory alignment is common when providing demonstrations for path/motion planning, and other time-series data.
- Dynamic Time Warping (DTW) aligns two sequences by warping the time axis iteratively until an optimal match between the two sequences is found.
  - DTW is a time series alignment algorithm developed originally for speech recognition.
  - Consider two trajectories (sequences of data points):

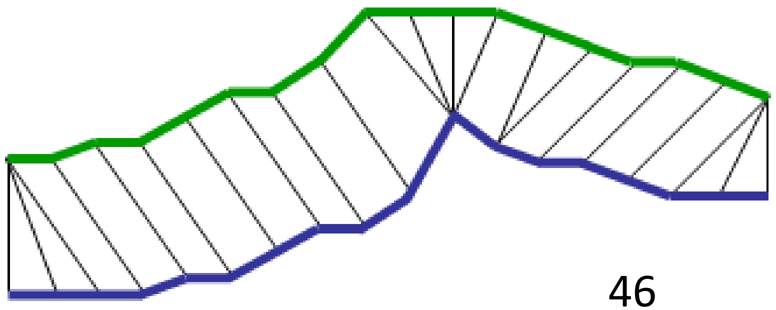
$$A = a_1, a_2, \ldots, a_i, \ldots, a_n$$
$$B = b_1, b_2, \ldots, b_j, \ldots, b_m$$

Sakoe,H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoust., Speech, and Signal Process., ASSP 26, 43-49 (1978).

Aligning Trajectories

- The two sequences are arranged on the sides of a grid, with one on the top and the other up the left-hand side.
- Both sequences start on the bottom left of the grid.
- Inside each cell a distance measure can be placed, comparing the corresponding elements of the two sequences.
- To find the best match or alignment between these two sequences, one need to find a path through the grid, which minimizes the total distance between them.
- This shortest path can be found using dynamic programming.

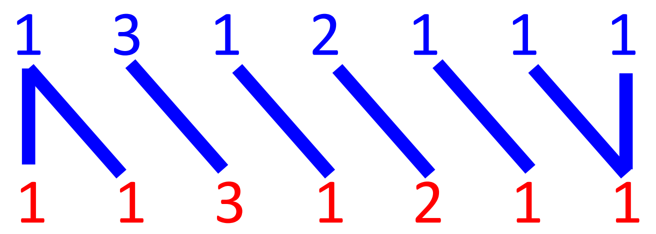




Aligning Trajectories

1 3 1 2 1 1 1

1 1 3 1 2 1 1



	1	3	1	2	1	1	1
1	0	8	0	3	0	0	0
1	0	8	0	3	0	0	0
2	3	5	3	0	3	3	3
1	0	8	0	3	0	0	0
3	8	0	8	5	8	8	8
1	0	8	0	3	0	0	0
1	0	8	0	3	0	0	0