Research statement — Sylvain Calinon, Idiap Research Institute

My work focuses on **human-centered robotics applications** in which the robots can **acquire new skills from only few demonstrations and interactions**. It requires the development of models that can **exploit the structure and geometry of the acquired data** in an efficient way, the development of optimal control techniques that can **exploit the learned task variations and coordination patterns**, and the development of intuitive interfaces to acquire meaningful demonstrations. The developed approaches can be applied to a wide range of manipulation skills, with robots that are either close to us (**assistive and industrial robots**), parts of us (**prosthetics and exoskeletons**), or far away from us (**teleoperation**).

This document presents an overview of the recent research developments in the *Robot Learning & Interaction group* at the Idiap Research Institute, with pointers to our most recent publications presenting the proposed approaches in further details. For each of the proposed research directions, I also discuss ongoing and further work that needs to be investigated to advance the proposed research lines.

Learning in a handful of trials

The field of machine learning has evolved toward approaches relying on huge amounts of data. In several application domains, these big datasets are already available or are inexpensive to collect/produce. In contrast, robotics is characterized by a different problem setting. It should instead be viewed as a **wide-ranging data problem**, with models that could start learning from small datasets, and that could still exploit more data if such data become available during the robot's lifespan. The current trend of machine learning relying on big datasets can bias the development of robot learning approaches in a negative way. In contrast to other fields, the data formats in robotics vary significantly across tasks, environments, users and platforms (different sensors and actuators). Then, the learned models often need to be interpretable to provide guarantees and to be linked with other techniques.

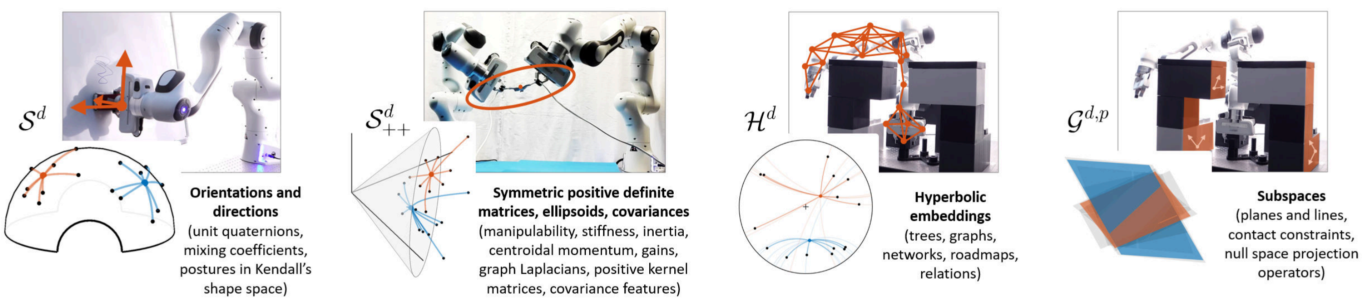
For these reasons, our group focuses on developing **robot learning approaches that can rely on only few demonstrations or trials**. The main challenge boils down to **finding structures that can be used in a wide range of tasks**, which requires us to advance research on several fronts, including (from low level to high level):

1. **Geometric structures**,
2. **Data structures**,
3. **Combination structures**,
4. **Learning structures**,

which are detailed in the next sections. Efficient robot skills acquisition requires the combination of these different forms of structures (at model and algorithm levels), as well as the consideration of an efficient trade-off between learning and exploitation.

In [6], we provide a survey of approaches in robotics to learn skills in a handful of trials.

1 Geometric structures



Robots act in a physical world. To speed up skills acquisition, the prior knowledge about this world can be embedded within the representations of skills and associated learning algorithms. This includes kinematics and dynamics properties such as the motion in a 3D space and the effect of gravity.

Data in robotics are characterized by simple but varied geometries. These geometric structures are often underexploited in learning, planning, control and perception. One of the most appealing use of **Riemannian geometry** in robotics is that it provides a principled and simple way to extend algorithms initially developed for Euclidean data to other manifolds, by efficiently taking into account prior geometric knowledge about these manifolds. In particular, Riemannian manifolds can be connected to a wide range of problems in robotics, including problems relying on Gaussian distributions. This includes techniques requiring **uncertainty and statistical modeling to be computed on structured non-Euclidean data**. Riemannian geometry can similarly be employed for a variety of data in robotics (see the above figure), including joint angles in revolving articulations, rigid body motions, unit quaternions to represent orientations, and symmetric positive definite matrices, which can represent sensory data processed as spatial covariances, inertia, stiffness/manipulability ellipsoids, as well as metrics used in the context of similarity measures.

The combination of statistics and Riemannian geometry offers many research opportunities, and can contribute to recent challenges in robotics. Further work can be organized in two categories. Firstly, the field of robotics is abundant in new research developments due to the interdisciplinary aspect and to the richness of the problems it involves. The common factor in many of these developments is that they rely on some form of statistics and/or propagation of uncertainty. These models and algorithms are typically developed for standard Euclidean spaces, where an extension to Riemannian manifolds has several benefits to offer.

Secondly, **several Riemannian manifolds remain largely underexploited in robotics**, despite the fact that most of them are mathematically well understood and characterized by simple closed-form expressions. **Grassmann manifolds** seem particularly promising to handle problems in robotics with high dimensional datapoints and only few training data, where subspaces are required in the computation to keep the most essential characteristics of the data. It is also promising for problems in which hierarchies are considered (such as inverse kinematics with kinematically redundant robots), because it provides a **geometric interpretation of nullspace structures**. Other Riemannian manifolds such as **hyperbolic manifolds** also seem propitious to bring a probabilistic treatment to dynamical systems, tree-based structures, graphs, Toeplitz/Hankel matrices or autoregressive models.

In [4], I review the use of Riemannian geometry in robot learning problems and present challenges that further need to be investigated.

Learning, tracking and transfer of manipulability ellipsoids

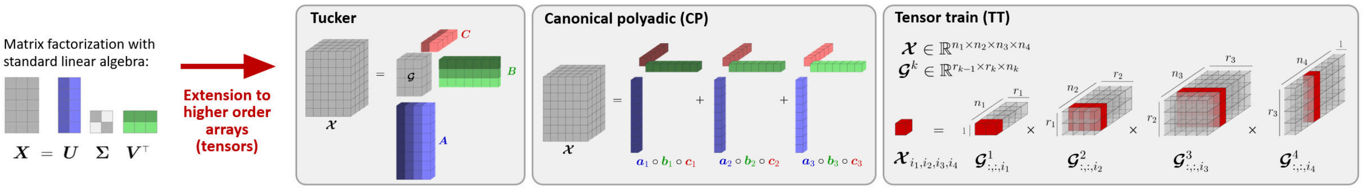
Skills transfer can exploit **stiffness and manipulability ellipsoids**, in the form of **geometric descriptors** representing the skills to be transferred to the robot. As these ellipsoids lie on **symmetric positive definite (SPD) manifolds**, Riemannian geometry can be used to learn and reproduce these descriptors in a probabilistic manner.

Manipulability ellipsoids come in different flavors. They can be defined at several points of interest, such as endeffectors (including tools held by the robot) and centers of mass. They can be defined for either positions or forces (including the orientation/rotational part). Manipulability can be described at either kinematic or dynamic levels, for either open or closed linkages (e.g., for bimanual manipulation or in-hand manipulation), and for either fully actuated or underactuated systems. This large set of manipulability ellipsoids provide **rich descriptors to characterize robot skills** for robots with legs and arms.

Manipulability ellipsoids can handle **skills transfer problems involving dissimilar kinematic chains**, such as transferring manipulation skills between humans and robots, or between two robots with different kinematic chains or capabilities. In such transfer problems, imitation at joint angle level is not possible due to the different structures, and imitation at endeffector(s) level is limited because it does not encapsulate postural information, which is often an essential aspect of the skill that needs to be transferred. Manipulability ellipsoids provide intermediate descriptors that allow postural information to be transferred indirectly, with the advantage that they allow different embodiments and capabilities to be considered in the skills transfer process.

In [7], we present an overview of our work on using manipulability ellipsoids for learning and control problems in robotics. The proposed approach can also be extended to other descriptors represented as ellipsoids, which further needs to be investigated. In particular, **stiffness, feedback gains, inertia, and centroidal momentum have similar structures**. The proposed approach could also potentially be extended to other forms of SPD matrices, such as kernel matrices used in machine learning algorithms to compute similarities, or graph graph Laplacians (a matrix representation of a graph that can for example be used to construct a low dimensional embedding of a graph).

2 Data structures



Another type of structure that we exploit relates to the organization of data as **multidimensional arrays** (also called tensors). These data appear in various robot tasks, either as the natural organization of sensory/motor data (tactile arrays, images, kinematic chains), as the result of standardized preprocessing steps (moving time windows, covariance features), data in multiple coordinate systems, or in the form of basis functions decompositions. Developed in the fields of **multilinear algebra** and **tensor methods**, these approaches extend linear factorization techniques such as singular value decomposition to multilinear decomposition, without requiring the transformation of the tensors to vectors or matrices. We exploit these techniques to provide robots with the capability **to learn tasks from only few tensor datapoints, by relying on the multidimensional structure of the data**.

Global optimization with tensor train decomposition

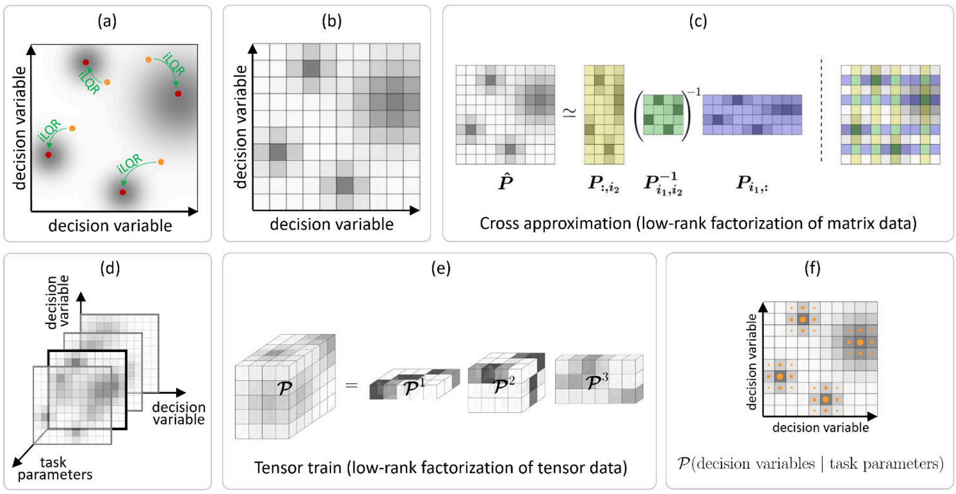


Figure 1: Combination of global and local optimization for fast adaptation to new situations, by leveraging offline and online computation. *(a)* Local optimization algorithms such as iLQR requires good initial estimates to speed up convergence, to reach better local minima and to find diverse solutions. The figure shows an illustration with two decision variables and four local minima of a cost function. *(b)* The decision variables space can be discretized to search for initial guesses near the local minima of the cost functions, which can be used to warm-start local optimizers such as iLQR. *(c)* The naïve solution consists of evaluating the cost for each set of decision variables, which typically does not scale well when more than 2 or 3 decision variables are used, which is the case in robot control problems. For 2D decision variables, a cross approximation algorithm can be used to approximate the cost function (treated as a probability distribution). The algorithm iteratively searches for row and column indices to reconstruct the full distribution from a sparse subset of rows and columns (depiced in colors). It also simultaneously estimates the number of rows and columns required, corresponding to the rank of the matrix. *(d)* Control problems in robotics have more than two dimensions, including both task parameters and decision variables. Learning the joint distribution of task parameters and decision variables can be conducted in an offline phase (through robot experiences and plays, possibly guided by human demonstrations). This offline learning phase is then followed by an online reproduction phase in which the robot needs to compute a controller as fast as possible, given a new set of task parameters describing a newly encountered situation. *(e)* The cross approximation algorithm can be extended to tensor data by exploiting tensor train decomposition. *(f)* This low-rank representation can be used for fast conditional sampling, allowing the robot to generate diverse controllers given a set of task parameters describing the situations and environment.

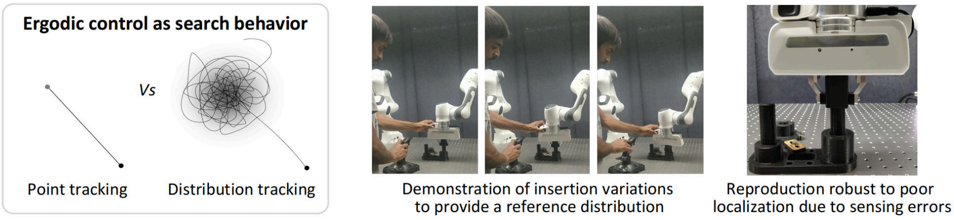
Learning and optimization problems in robotics are characterized by two types of variables: 1) **task parameters** representing the situation that the robot encounters, typically related to environment variables such as locations of objects, users or obstacles; and 2) **decision variables** related to actions that the robot takes, typically related to a controller acting within a given time window, or the use of basis functions to describe trajectories in control or state spaces. For each change of task parameters, decision variables need to be recomputed as fast as possible, so that the robot can fluently collaborate with users and can swiftly react to changes in its environment.

Within the MEMMO and LEARN-REAL projects, we investigate the roles of offline and online learning optimization to attain such objective. The problem is formalized as an **optimal control problem** with a cost function to minimize, parameterized by task parameters and decision variables. We investigate the use of **tensor train (TT) decomposition as a model to learn the structure between the task parameters, the decision variables and the**

resulting cost expressed in the form of a probability distribution, which then allows solutions to be sampled from a conditional distribution by TT-cross approximation. We exploit this structure to gather prior knowledge in an offline phase, which is further used for fast online decision making, with local Gauss-Newton optimization in the form of **iterative linear quadratic regulators (iLQR)**, see Figure 1 for an overview.

In [11], we demonstrated the capability of the approach for trajectory optimization within a varied set of control and planning problems with robot manipulators.

Ergodic control applied to insertion tasks



A conventional tracking problem in robotics is characterized by a target to reach, requiring a controller to be computed to reach this target. In **ergodic control**, instead of providing a single target point, a probability distribution is given to the robot, which must cover the distribution in an efficient way. Ergodic control thus consists of moving within a spatial distribution by spending time in each part of the distribution in proportion to its density (namely, **"tracking a distribution" instead of "tracking a point"**). The resulting controller generates **natural exploration behaviors**. The underlying objective takes a simple form, corresponding to a tracking problem in the spectral domain (tracking of frequency components). The advantage of such a control formulation is that it can be easily combined with other control objectives and constraints.

In robotics, the approach can be exploited in a wide range of problems requiring the **automatic exploration of regions of interest**. This is particularly helpful when the available sensing information is not accurate enough to fulfill the task with a standard controller, but where this information can still guide the robot towards promising areas. In a collaborative task, it can also be used when the operator's input is not accurate enough to fully reproduce the task, which then requires the robot to explore around the requested input (e.g., a point of interest selected by the operator). For picking and insertion problems, ergodic control can be applied to move around the picking/insertion point, thereby facilitating the prehension/insertion. It can also be employed for active sensing and localization (either detected autonomously, or with help by the operator). Here, the robot can plan movements based on the current information density, and can recompute the commands when new measurements are available (i.e., updating the spatial distribution used as target).

In ergodic control, the problem is originally formulated as a **spectral multiscale coverage** (SMC) objective. It requires the spatial distribution to be decomposed as Fourier series, with a cost function comparing the spectral decomposition of the robot path with the spectral decomposition of the distribution. The resulting controller allows the robot to explore the given spatial distribution in a natural manner, **by starting from a crude exploration and by refining the search progressively** (i.e., matching the Fourier coefficients with an increasing importance from low to high frequency components).

The downside of the original ergodic control approach is that it does not scale well to problems requiring exploration in search space of more than two dimensions. In particular, the original approach would be too slow to consider full distributions in 6D spaces, which would be ideally required. Indeed, both position and orientation of endeffector(s) matter in most robot problems, including manipulation, insertion, welding, or the use of tools at the robot endeffectors.

In [12], within the COLLABORATE European project, we demonstrated that the original problem formulation can be conserved **by efficiently compressing the Fourier series decomposition with tensor train (TT) factorization**. The proposed solution is efficient both computationally and storage-wise, hence making it suitable for online implementations, as well as to tackle robot learning problems with a low quantity of training data. The above figure shows an overview of an insertion experiment conducted with the Siemens gears benchmark requiring full 6D endeffector poses. Further work is required to extend the approach to online active sensing applications in which the distributions can change based on the data collected by ergodic control.

3 Combination structures

The term **movement primitives** refers to an organization of continuous movements in the form of a superposition in parallel and in series of simpler signals, which can be viewed as "building blocks" to create more complex movements.

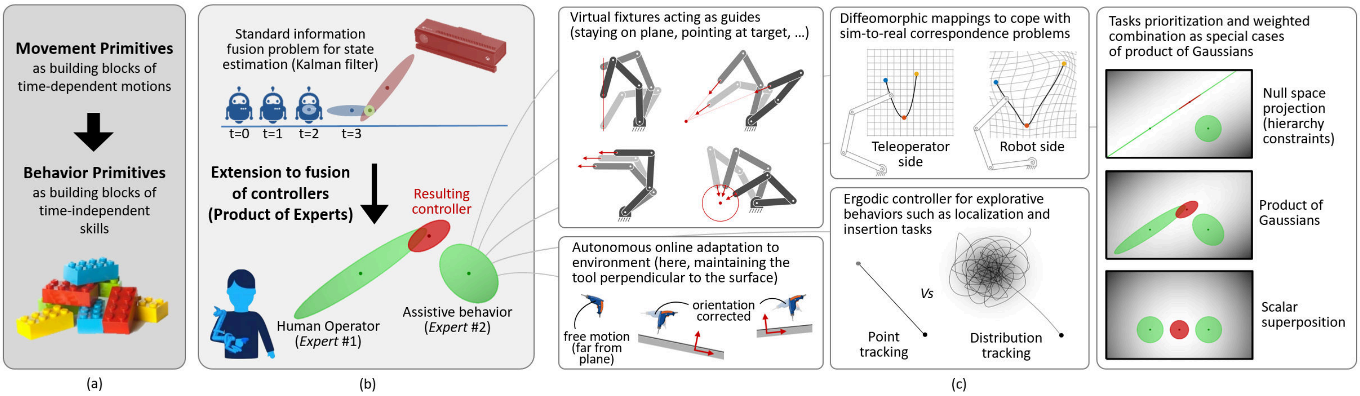


Figure 2: Combination of robot controllers as a product of experts. *(a)* We extend the principle of movement primitives (superposition of basis functions as time-dependent trajectories) to behavior primitives (superposition of time-independent controllers). *(b)* State estimation in robotics is typically formulated as an information fusion problem. We propose to combine controllers in the same principled way by using a product of experts (PoE) formulation in which the user is part of the shared control problem, with the other controllers assisting the user to achieve the task. *(c)* We develop behavior primitives in both position and force domains, including the autonomous adaptation to object locations, the consideration of tasks variations and tasks prioritization, or the use of ergodic controllers for exploration behaviors.

This principle of **superposing high-level "bricks of motion" from a dictionary**, coined in the context of motor control, remains valid for a wide range of continuous time signals (for both analysis and synthesis). In particular, this notion can be extended to *behavior primitives*, which form a richer set of assistive behaviors, see Figure 2-**(a)**. These behavior primitives correspond to controllers that are either myopic or anticipative, with either time-independent or time-dependent formulations.

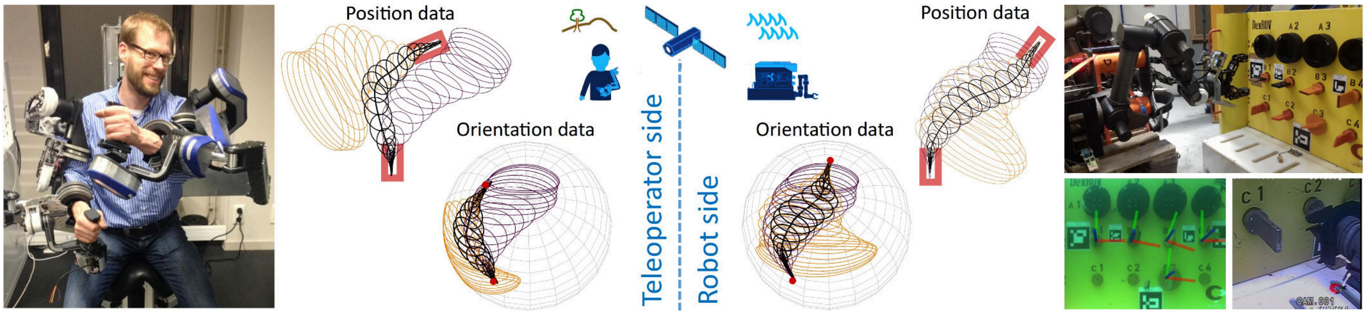
In [10], we showed that the combination of behavior primitives can be formalized as an **information fusion problem** in which several sources of information can be modeled as probability distributions. This results in a **product of experts (PoE)**, a machine learning technique modeling a probability distribution by combining the output from several simpler distributions. The core idea is **to combine several distributions** (called experts) by multiplying their density functions. **This allows each expert to make decisions on the basis of a few dimensions, without having to cover the full dimensionality of a problem**. A PoE corresponds to an "and" operation, which contrasts with a mixture model that corresponds to an "or" operation (by combining several probability distributions as a weighted sum of their density functions). Thus, each component in a PoE represents a soft constraint. For an event to be likely under a product model, all constraints must be (approximately) satisfied. In contrast, in a mixture model, an event is likely if it matches (approximately) with any single expert.

With Gaussian distributions, the fusion problem simplifies to a **product of Gaussians (PoG)**, which can be solved analytically, where the distributions can either represent robot commands at the current time step (myopic control system), or trajectory distributions in the control space (anticipative planning system). State estimation is an example of problems classically solved as an information fusion problem, resulting in a PoG that takes into account uncertainty in motion and sensor(s) models (a well-known example is the Kalman filter), see Figure 2-*(b)*. A control problem in a collaborative robot task can be treated in a similar mathematical framework by relying on a product of experts (PoE). This approach allows multiple behavior primitives to be combined in parallel (including prioritization with nullspace projection structures, see [10] for details). In this framework, each operator (human or virtual) contributes to the task by taking into account the (co)variations allowed in a task and the uncertainty of the expertise.

The combination of behavior primitives can be treated within a similar mathematical framework, by relying on products of experts, where **each expert takes care of a specific aspect of the task to achieve**. For robot applications, this approach allows the combination of different controllers, which can be learned separately or altogether (by variational inference, see [10] for details). With this formulation, a robot can counteract perturbations that have an impact on the fulfillment of the task, while ignoring other perturbations. It also allows bridges to be created with research in biomechanics and motor control, with formulations including **minimal intervention principles**, **uncontrolled manifolds** or **optimal feedback control**. This formulation also opens new perspectives to tackle challenges in manipulation and assembly with either autonomous robots or cobots, including the handling of both force and position information, as well as the simultaneous handling of multiple objectives (including physical compliance, manipulability, safety, and task hierarchies). Figure 2-*(c)* presents examples of assistive behaviors in the context of manipulation and assembly skills.

Next, I show how the PoE/PoG principle can be used for fusing information from multiple coordinate systems.

Task-parameterized movements



To facilitate the acquisition of manipulation skills, **task-parameterized models** can be exploited to take into account that movements typically relate to objects, tools or landmarks in the robot's workspace.

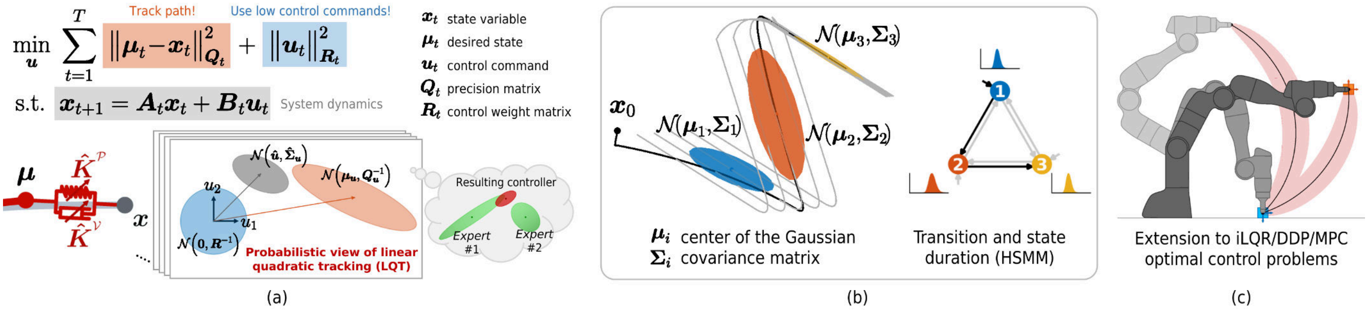
In [1], I present a review of task-parameterized models for learning and control applications. The approach consists of **encoding a movement in multiple coordinate systems** (namely, from the perspectives of different objects), in the form of trajectory distributions. In a new situation (i.e., for new object locations), the reproduction problem corresponds to a fusion problem, where the variations in the different coordinate systems are exploited to generate a movement reference tracked with varying gains, providing the robot with a **variable impedance behavior that automatically adapts the precision** required by the different phases of the task. For example, in a pick-and-place task, the robot will be stiff if the object needs to be reached/dropped in a precise way, and will remain compliant in the other parts of the task. We also showed that the approach could be extended to task prioritization with nullspace projection operators [13].

For user assistance in collaborative operations, this approach can also be used for **local adaptation to objects or landmarks in the robot's environment**, such as automatically correcting the orientation of a tool held by the operator to ensure it remains perpendicular to a surface. The above figure presents an example of application for remote valve turning operations, developed in the context of the DexROV European project. In this example, a **task-parameterized Gaussian mixture model (TP-GMM)** approach is used within an optimal control approach to teleoperate an underwater robot from distance, with a teleoperator wearing an exoskeleton and visualizing a copy of the robot workspace in a virtual environment. Because of the long communication delays between the teleoperator and the robot, the locations of the objects or tools of interest are not the same on the teleoperator side and on the robot side. With a parameterization associated with the locations of objects and tools, this discrepancy can be handled by adapting locally the movement representation to the position and orientation of the objects/tools, represented as coordinate systems. The central part of the above figure depicts two coordinate systems (with models represented in orange and purple), corresponding respectively to the robot and to the valve to be turned. A motion relative to the valve and to the robot is encoded as GMMs in the two respective coordinate systems. During teleoperation, the distributions are rotated and translated according to the current situations on the teleoperator side and on the robot side. Products of Gaussians are then computed at each side to fuse these representations.

The task-parameterized principle can be used as a first starting point to study the fusion of more elaborated coordinate systems. In the above figure, Cartesian coordinate systems centered on several points of interest were considered. We currently investigate extensions of the approach to a wider range of coordinate systems, in the form of **manifolds taking into account object symmetries and objects affordances** (incl. cylindrical and spherical coordinate systems), see [15] for a first work in this direction.

Next, I show that the description of movements as trajectory distributions can be extended to control problems by adopting optimal control formulations, and that such a mathematical framework can be used to create bridges with research in imitation.

Probabilistic optimal control



Skill acquisition encompasses a **broad spectrum of learning approaches, from action-level imitation to goal-level emulation**, see [2] for a short review. While emulation requires higher cognitive skills to infer the intended goals behind a set of actions, imitation allows us to acquire skills in a simple but limited way, by directly mimicking the movements and actions shown to the learner. Because of its higher level nature, emulation can easily be misinterpreted as being the only useful mechanism. The two are instead complementary, in the sense that emulation allows greater generalization (i.e., the learner understands the purpose of the task and can reproduce it in possibly different ways), while imitation allows us to acquire tasks fast, even if all underlying goals are not fully understood (**"copy all, refine later" strategy**).

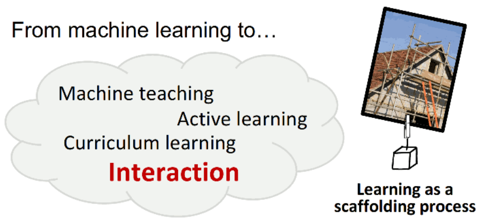
Similarly, within engineering applications, research on optimal control and reinforcement learning (RL) is tightly connected to the emulation approach, while research on movement primitives is tightly connected to the imitation approach. The mutual benefit can be illustrated by considering emulation as a way to make the system more generalizable, while imitation/mimicking can improve the learning rate and the learning scope. Our work investigates the use of **optimal control** as a formulation based on a cost that implements the **goal-level nature of emulation**, that we combine with the use of **control primitives** as a formulation based on basis functions to implement the **action-level nature of imitation**. In [3], I present several forms of basis functions can be considered to build such control primitives, including radial basis functions (RBFs), Bernstein polynomials, and Fourier series.

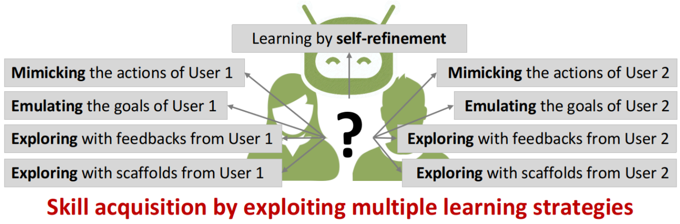
As an example of implementation of the above principle presented in [5], we can rely on **Linear quadratic tracking (LQT)** as a simple form of optimal control that trades off tracking and control costs expressed as quadratic terms over a time horizon, with the evolution of the state described in a linear form. This constrained problem can be solved by expressing the state and the control commands as trajectories, corresponding to a least squares solution. A **probabilistic interpretation of the LQT solution** can be built by using the residuals of this estimate, see (a) in the above figure. This approach allows the **creation of bridges between learning and control**.

For example, in learning from demonstration, the observed (co)variations in a task can be formulated as an LQT objective function, see (b) in the above figure. It then provides a **trajectory distribution in control space** that can be converted to a trajectory distribution in state space, with all operations being analytic. Thus, the LQT formulation allows a controller to be computed from a high-level representation formalized as a cost function, which allows the approach to be combined easily with other optimal control strategies. Notably, it allows multiple demonstrations of a movement to be used to estimate a feedback controller. As a result, the **(co)variations in the demonstrations** are exploited to provide a **minimal intervention control** strategy that will **selectively reject perturbations**, based on the impact that they can have on the task to achieve. This is effectively attained by automatically regulating the gains in accordance to the variations in the demonstrations, with low gains in parts of the movement allowing variations, and higher gains for parts of the movement that are invariant in the demonstrations.

While simple LQT already highlights the advantage of the proposed approach, further work is required to extend it to **model predictive control (MPC), iterative LQR (iLQR) and differential dynamic programming (DDP)**, with probabilistic solutions that need this time to be interpreted locally at each iteration step of the algorithm, see (c) in the above figure and [9] for a first work in this direction.

4 Learning structures





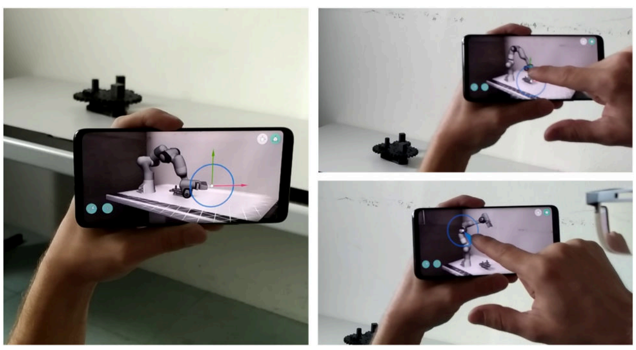
So far, I presented structures that were taking the form of algorithms and/or representations. To reduce the amount of required data, an additional opportunity that we can seize is that machine learning in robotics goes beyond the standard training-set and testing-set paradigm. Indeed, we can exploit a number of interactive learning mechanisms to acquire/generate better data on-the-spot, including **active learning**, **machine teaching** (by generating data to train our robots), **curriculum learning** (by providing data of increased complexity that adapt to the learner), and **bilateral interactions** that rely on several social mechanisms to transfer skills more efficiently, see the above figure for an illustration. Indeed, **skills acquisition in robotics is a scaffolding process rather than a standard learning process**. In this scaffolding metaphor, the robot needs a lot of structures at the beginning, which can then be progressively dismantled when the robot progresses: there is a continuous evolution from full assistance to full autonomy.

To acquire manipulation skills, **humans greatly benefit from the combination of several learning modalities**. Similarly, multiple learning strategies could jointly be exploited to transfer skills to robots, while facilitating the evaluation of the robot's current capability at executing the task. The research direction of **orchestrating different learning modalities** is still in its infancy in robotics research. However, this research direction has the potential to improve the robustness of the acquired manipulation and assembly skills, by allowing diverse forms of constraints to be considered (at the level of the users and the environments), as well as an improved monitoring of the robot knowledge (by testing the acquired skills in diverse situations).

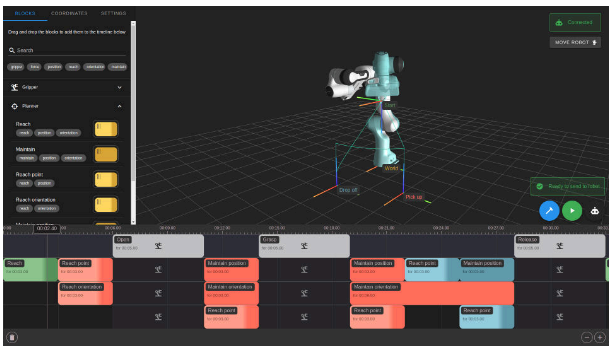
The other important advantage is that such a combination of learning strategies allows each individual learning modality to be simplified. Indeed, skills acquisition with a single learning strategy can be unnecessarily complex, and similarly to us, robots might not be able to acquire skills efficiently by using a single learning modality. Thus, instead of focusing on the improvement of algorithms and models for a specific learning strategy, robotics would highly benefit from further research to investigate the **meta-learning problem of combining efficiently existing learning methods, without defining the sequence and/or organization in advance**.

In [8], we addressed this question with a first attempt at combining social and intrinsically-motivated Learning, but further study is still required to generalize the approach to more diverse learning strategies.

User interfaces for robot programming



Portable augmented reality interface



Timeline GUI for robot programming

When a robot acquires skills by user guidance, an associated challenge is to develop **user-friendly interfaces for the users to teach and monitor the acquisition of robot manipulation skills**. First, this is important to let users provide useful demonstrations of the task, by allowing them to concentrate on the skill to transfer instead of nonessential hindering factors. This is also important to let them assess the current capability of the robot, in order to provide demonstrations or corrections that are the most appropriate with respect to the current capability of the robot. Thus, **interfaces enabling bidirectional interactions** need to be developed.

Currently, the most commonly used interfaces to teach robots new tasks in industrial applications are the **teaching**

pendants. Recent torque-controlled or collaborative robot manipulators bring new programming interaction capabilities, by letting operators drive the robots manually, with backdrivable motors and controllers that compensate for the effect of gravity. The resulting interfaces allow users to move robots freely in space as if there was no motorization in the articulations, and as if the robots had no weight. **This kinesthetic teaching capability can be extended to more elaborated forms of virtual fixtures** by exploiting impedance through the torque control capability of the robot, which provides a physical guidance capability to facilitate manipulation skill transfer, see (c) in Figure 2.

Research in **virtual reality (VR), augmented reality (AR) and mixed reality (MR)**, also gathered under the name of **extended reality (XR)**, offer promising alternative approaches to interact with robots, for both teaching and monitoring aspects. A subset of these approaches rely on the AR capability of recent smartphones or tablets, by displaying on the screen the video stream from the cameras on the back of the devices, and by superposing to these images virtual robots and/or virtual guides (typically, paths, labels or coordinate systems). The device then appears as a transparent window that users can freely move to observe a virtual scene superposed to the real scene, by changing the angle of view or by moving closer/further away, which provides a very intuitive way of analyzing existing movements and programming new robot movements. Such an interface allows users to inspect a taught path, for example **by replaying consecutively a robot motion on the virtual robot before executing it on the real robot**.

In [14], we developed and tested an AR interface for smartphones in the context of programming industrial (de)insertion tasks with a NIST Task Board, a benchmark platform used to evaluate assembly performance, see above figure.

Another line of work that we currently pursue relates to the **edition and visualization of costs functions in optimal control problems** (in particular, using an iLQR formulation). The approach is investigated in the context of robot manipulation tasks, which typically require cost functions composed of different objectives organized both in series and in parallel. Because of this organization, we study the exploitation of **graphical user interfaces in the form of timelines**, with blocks and keyframes that can be (re)organized, similarly to the process of editing a video or an animation, see above figure. Such an approach differs from the standard programming interfaces based on drag-and-drop blocks that often present the task in a sequential manner (typically, with a top-down organization of blocks), but that do not allow multiple processes to run in parallel. Further work is required to evaluate these two programming paradigms and develop interfaces that could potentially gather the two types of interfaces.

References

[1] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, January 2016.

[2] S. Calinon. Learning from demonstration (programming by demonstration). In M. H. Ang, O. Khatib, and B. Siciliano, editors, *Encyclopedia of Robotics*. Springer, 2019.

[3] S. Calinon. Mixture models for the analysis, edition, and synthesis of continuous time series. In N. Bouguila and W. Fan, editors, *Mixture Models and Applications*, pages 39–57. Springer, 2019.

[4] S. Calinon. Gaussians on Riemannian manifolds: Applications for robot learning and adaptive control. *IEEE Robotics and Automation Magazine (RAM)*, 27(2):33–45, June 2020.

[5] S. Calinon and D. Lee. Learning control. In P. Vadakkepat and A. Goswami, editors, *Humanoid Robotics: a Reference*, pages 1261–1312. Springer, 2019.

[6] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Trans. on Robotics*, 32(2):328–347, Apr 2020.

[7] N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon. Geometry-aware manipulability learning, tracking and transfer. *International Journal of Robotics Research (IJRR)*, 40(2–3):624–650, 2021.

[8] T. Kulak and S. Calinon. Combining social and intrinsically-motivated learning for multi-task robot skill acquisition. *IEEE Trans. on Cognitive and Developmental Systems*, 2021.

[9] T. S. Lembono and S. Calinon. Probabilistic iterative LQR for short time horizon MPC. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 556–562, 2021.

[10] E. Pignat, J. Silvério, and S. Calinon. Learning from demonstration using products of experts: Applications to manipulation and task prioritization. *International Journal of Robotics Research (IJRR)*, 41(2):163–188, 2022.

[11] S. Shetty, T. Lembono, T. Löw, and S. Calinon. Tensor train for global optimization problems in robotics. *arXiv:2206.05077*, 2022.

[12] S. Shetty, J. Silvério, and S. Calinon. Ergodic exploration using tensor train: Applications in insertion tasks. *IEEE Trans. on Robotics*, 38(2):906–921, 2022.

[13] J. Silvério, S. Calinon, L. Rozo, and D. G. Caldwell. Learning task priorities from demonstrations. *IEEE Trans. on Robotics*, 35(1):78–94, 2019.

[14] Y. J. Thoo, J. Maceiras, P. Abbet, M. Racca, H. Girgin, and S. Calinon. Online and offline robot programming via augmented reality workspaces. *arXiv:2107.01884*, pages 1–8, 2021.

[15] B. Ti, Y. Gao, J. Zhao, and S. Calinon. Imitation of manipulation skills using multiple geometries. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2022.