Stochastic learning and control in multiple coordinate systems

Sylvain Calinon*

***Abstract*— A probabilistic interpretation of model predictive control is presented, enabling extensions to multiple coordinate systems. The resulting controller follows a minimal intervention principle, by learning and retrieving movements through the coordination of several frames of reference. When combined with a generative model, the approach can be used in various human-robot applications that are discussed in the paper.**

I. Introduction

Model predictive control (MPC) encompasses a wide range of techniques to control a robot with the ability to anticipate future events. This is achieved by optimizing a cost function over a finite time horizon, by implementing the current timeslot, and by reiterating the operation at desired time intervals. Linear MPC is used in the majority of applications, where the system to control is defined in a linear form (often through linearization over the operating range), with the underlying feedback mechanism compensating for prediction errors due to the mismatch between the model and the process.

MPC is well known in control, but learning applications can also benefit from the formalism of MPC and the associated algorithms. In particular, the most basic form of MPC with a cost function composed of a sum of quadratic terms has a representation that is linked to problems in statistics, formulated as log-likelihoods of Gaussian distributions. Indeed, solving a problem whose cost is a sum of quadratic terms can alternatively be treated as a product of linearly transformed Gaussians (see Appendix I), whose result is a Gaussian with a center and covariance corresponding to the average solution and an error that can be used as information to determine how much we can move from the average while still fulfilling the task constraints, see [1] for details.

From a learning perspective, a particularly interesting feature of MPC is that it can exploit the covariance information contained in several examples of a movement to find a controller achieving a minimal intervention strategy [2]. The retrieved control commands comprise feedforward and feedback components corresponding to a coordinated tracking behavior with gains that can vary during the task.

For transferring skills through human-robot collaboration, the combination of MPC with generative models in machine learning provides a framework to handle both analysis and synthesis challenges. Indeed, the controller will reflect the observed correlational structures in the samples. This is for example important in learning from demonstration, where an efficient transfer of skills is not characterized by the accurate reproduction of a trajectory, but instead requires the robot to take several sources of variations into account, including the variations of the task (e.g., multiple options or redundancy that do not influence task performance).

For skills transfer between living and artificial systems (from a human to a robot, from a robot to another robot, or from an internal simulator to the real robot), it is primordial to take into account that *being skillful* does not mean *being accurate*. Being skillful requires the system to exploit task (co)variations so that perturbations can be rejected efficiently. Precision is only a special case of skill requirements, which is in practice usually required only in some parts of an overall task, and which does not necessary apply to all task variables (a task can require precision in some directions and be loose in some others). Such constraints can often be better described as coordination/synergy requirements instead of considering each control variable individually.

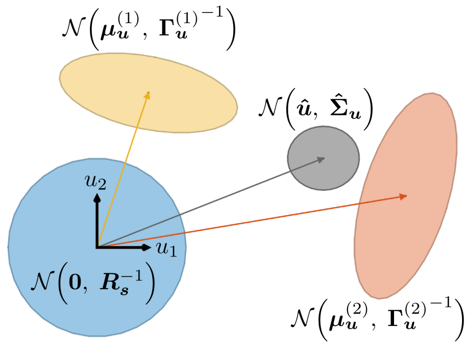

Fig. 1: Retrieval of control commands in a probabilistic form by interpreting MPC as a product of Gaussians, and extending it to the consideration of multiple coordinate systems that are simultaneously exploited for the reproduction of the task. The graph depicts only the first two dimensions of control commands organized in a lifted vector form of size $\mathbb{R}^{DT}$, see also Table I. The blue Gaussian corresponds to the control cost, isotropic and centered at zero (no acceleration is optimal for this part of the cost). The orange and red Gaussians correspond to control commands tracking desired targets expressed in two different coordinate systems. Each center depicts the optimal command for the respective frame, with a covariance depicting the required precision. The product of the three Gaussians yields another Gaussian depicted in gray, representing the estimated command, where the center is the set of control commands solving the complete cost function, and the covariance depicts the potential (co)variations around this estimate. The result is a trade-off between acceleration and tracking accuracy in multiple frames of reference.

*Idiap Research Institute, CH-1920 Martigny, Switzerland. E-mail: sylvain.calinon@idiap.ch. This work was in part supported by the Swiss National Science Foundation through the I-DRESS project (CHIST-ERA), and by the DexROV Project through the EC Horizon 2020 programme (Grant #635491).

Sym.	Function	Dim.
D	Dimension of position	$\mathbb{N}$
C	Order of the controller	$\mathbb{N}$
T	Horizon (nb of time steps)	$\mathbb{N}$
$\boldsymbol{s}$	$\{s_1, s_2, \ldots, s_T\}$	$\mathbb{N}^T$
At time step t:		
s_t	State identifier	$\mathbb{N}$
$\boldsymbol{x}_t$	Position	$\mathbb{R}^D$
$\dot{\boldsymbol{x}}_t$	Velocity	$\mathbb{R}^D$
$\boldsymbol{\zeta}_t$	State ($\big[\boldsymbol{x}_t^\top, \dot{\boldsymbol{x}}_t^\top\big]^\top$ for $C\!=\!2$)	$\mathbb{R}^{DC}$
$\boldsymbol{\mu}_t$	Desired target state	$\mathbb{R}^{DC}$
$\boldsymbol{u}_t$	Control command	$\mathbb{R}^D$
$\boldsymbol{Q}_t$	State tracking weights	$\mathbb{R}^{DC\times DC}$
$\boldsymbol{R}_t$	Control weights	$\mathbb{R}^{D\times D}$
In lifted vector/matrix form:		
$\boldsymbol{\zeta}$	$\big[\boldsymbol{\zeta}_1^\top, \boldsymbol{\zeta}_2^\top, \ldots, \boldsymbol{\zeta}_T^\top\big]^\top$	$\mathbb{R}^{DCT}$
$\boldsymbol{u}$	$\big[\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, \ldots, \boldsymbol{u}_T^\top\big]^\top$	$\mathbb{R}^{DT}$
$\boldsymbol{\mu}_{\boldsymbol{s}}$	$\big[\hat{\boldsymbol{\zeta}}_{s_1}^\top, \hat{\boldsymbol{\zeta}}_{s_2}^\top, \ldots, \hat{\boldsymbol{\zeta}}_{s_T}^\top\big]^\top$	$\mathbb{R}^{DCT}$
$\boldsymbol{Q}_{\boldsymbol{s}}$	blockdiag$(\boldsymbol{Q}_{s_1}, \boldsymbol{Q}_{s_2}, \ldots, \boldsymbol{Q}_{s_T})$	$\mathbb{R}^{DCT\times DCT}$
$\boldsymbol{R}_{\boldsymbol{s}}$	blockdiag$(\boldsymbol{R}_{s_1}, \boldsymbol{R}_{s_2}, \ldots, \boldsymbol{R}_{s_T})$	$\mathbb{R}^{DT\times DT}$
Transfer matrices:		
$\boldsymbol{A}_t$	State matrix	$\mathbb{R}^{DC\times DC}$
$\boldsymbol{B}_t$	Input matrix	$\mathbb{R}^{DC\times D}$
$\boldsymbol{S}_{\boldsymbol{\zeta}}$	Lifted state matrix	$\mathbb{R}^{DCT\times DC}$
$\boldsymbol{S}_{\boldsymbol{u}}$	Lifted input matrix	$\mathbb{R}^{DCT\times DT}$
Normal distributions:		
$\mathcal{N}\Big(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\Big)$	Control commands	$\mathbb{R}^{DT}$
$\mathcal{N}\Big(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\zeta}}\Big)$	Reconstructed states	$\mathbb{R}^{DCT}$
$\mathcal{N}\Big(\boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{\Gamma}_{\boldsymbol{u}}^{-1}\Big)$	Intermediary variables	$\mathbb{R}^{DT}$

TABLE I: Symbols, functions and dimensions used to describe the proposed approach.

II. A PROBABILISTIC INTERPRETATION OF MPC

A linear MPC problem with a cost function in a quadratic form can be treated in a probabilistic manner with standard linear algebra by exploiting two properties of multivariate normal distributions: linear combination and product properties. Table I will be used as notation reference to describe the proposed approach. Uppercase letters in normal typeface describe dimensions. Bold uppercase letters describe matrices. Bold lowercase letters describe vectors.

The problem is that of estimating a controller $\boldsymbol{u}_t$ for the discrete linear dynamical system

$$\boldsymbol{\zeta}_{t+1} = \boldsymbol{A}_t\boldsymbol{\zeta}_t + \boldsymbol{B}_t\boldsymbol{u}_t, \tag{1}$$

with state variable $\boldsymbol{\zeta}_t = \big[\boldsymbol{x}_t^\top, \dot{\boldsymbol{x}}_t^\top\big]^\top \in\mathbb{R}^{DC}$ (we use here $C=2$, corresponding to a state variable up to the first derivative), and a position $\boldsymbol{x}_t$ described by a vector of dimension D. The problem is formulated as the minimization of the cost

$$\begin{aligned} c &= \sum_{t=1}^{T} \Big((\boldsymbol{\mu}_t-\boldsymbol{\zeta}_t)^\top \boldsymbol{Q}_t (\boldsymbol{\mu}_t-\boldsymbol{\zeta}_t) \;+\; \boldsymbol{u}_t^\top \boldsymbol{R}_t\, \boldsymbol{u}_t \Big) \\ &= (\boldsymbol{\mu}_{\boldsymbol{s}} - \boldsymbol{\zeta})^\top \boldsymbol{Q}_{\boldsymbol{s}} (\boldsymbol{\mu}_{\boldsymbol{s}} - \boldsymbol{\zeta}) \;+\; \boldsymbol{u}^\top \boldsymbol{R}_{\boldsymbol{s}} \boldsymbol{u}, \end{aligned} \tag{2}$$

with $\boldsymbol{\zeta}$, $\boldsymbol{u}$ and $\boldsymbol{\mu}_{\boldsymbol{s}}$ describing respectively the evolution of the state, control and target variables for a time window of size T, organized in a lifted vector form, see Table I for details. $\boldsymbol{Q}_{\boldsymbol{s}}$ and $\boldsymbol{R}_{\boldsymbol{s}}$ represent the evolution of the required tracking precision and cost on the control inputs. A linear unconstrained MPC problem can be solved with simple linear algebra, by expressing all future states $\boldsymbol{\zeta}_t$ as an explicit function of the state $\boldsymbol{\zeta}_1$. In a matrix form, we get

$$\boldsymbol{\zeta} = \boldsymbol{S}_{\boldsymbol{\zeta}}\boldsymbol{\zeta}_1 + \boldsymbol{S}_{\boldsymbol{u}}\boldsymbol{u}, \tag{3}$$

where $\boldsymbol{S}_{\boldsymbol{\zeta}}$ and $\boldsymbol{S}_{\boldsymbol{u}}$ are transfer matrices, see Appendix II for details. Substituting (3) into (2), we get the cost function

$$c = (\boldsymbol{\mu}_{\boldsymbol{u}} - \boldsymbol{u})^\top \boldsymbol{\Gamma}_{\boldsymbol{u}} (\boldsymbol{\mu}_{\boldsymbol{u}} - \boldsymbol{u}) + \boldsymbol{u}^\top \boldsymbol{R}_{\boldsymbol{s}}\boldsymbol{u}, \tag{4}$$

where $\boldsymbol{\mu}_{\boldsymbol{u}} = \tilde{\boldsymbol{S}}_{\boldsymbol{u}}(\boldsymbol{\mu}_{\boldsymbol{s}} - \boldsymbol{S}_{\boldsymbol{\zeta}}\boldsymbol{\zeta}_1)$ and $\boldsymbol{\Gamma}_{\boldsymbol{u}} = \boldsymbol{S}_{\boldsymbol{u}}^\top\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{S}_{\boldsymbol{u}}$ are introduced for notation convenience, with $\tilde{\boldsymbol{S}}_{\boldsymbol{u}}$ defined for commodity so that $\boldsymbol{S}_{\boldsymbol{u}}\tilde{\boldsymbol{S}}_{\boldsymbol{u}} = \boldsymbol{I}$ (no explicit computation is required).

Because c in (4) can be expressed as a weighted sum of quadratic error terms, the minimization of (4) corresponds to the product of Gaussians (see Appendix I)

$$\begin{aligned} \mathcal{N}\Big(\hat{\boldsymbol{u}},\, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\Big) &\sim \mathcal{N}\Big(\boldsymbol{\mu}_{\boldsymbol{u}},\, \boldsymbol{\Gamma}_{\boldsymbol{u}}^{-1}\Big)\, \mathcal{N}\Big(\boldsymbol{0},\, \boldsymbol{R}_{\boldsymbol{s}}^{-1}\Big), \\ \text{with}\quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} &= (\boldsymbol{\Gamma}_{\boldsymbol{u}} + \boldsymbol{R}_{\boldsymbol{s}})^{-1} \\ &= (\boldsymbol{S}_{\boldsymbol{u}}^\top\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{S}_{\boldsymbol{u}} + \boldsymbol{R}_{\boldsymbol{s}})^{-1}, \\ \hat{\boldsymbol{u}} &= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\boldsymbol{\Gamma}_{\boldsymbol{u}}\boldsymbol{\mu}_{\boldsymbol{u}} \\ &= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\boldsymbol{S}_{\boldsymbol{u}}^\top\boldsymbol{Q}_{\boldsymbol{s}}\, (\boldsymbol{\mu}_{\boldsymbol{s}} - \boldsymbol{S}_{\boldsymbol{\zeta}}\boldsymbol{\zeta}_1). \end{aligned} \tag{5}$$

Fig. 1 illustrates the process. The product in (5) shows the role of $\boldsymbol{R}_{\boldsymbol{s}}$ as a regularization term in the process (prior information defining the importance of having a low cumulated sum of acceleration commands).

Alternatively to the above computation, differentiating with respect to $\boldsymbol{u}$ and equating to zero provide the same sequence of control inputs $\hat{\boldsymbol{u}}$, corresponding to a weighted least squares estimate with Tikhonov regularization (ridge regression). The error on the ridge regression estimate can then be used to compute a covariance $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}$ in control space.

By using the linear relation in (3) and the linear transformation property of Gaussians (see Appendix I), the distribution $\mathcal{N}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}})$ in control space can optionally be converted to a distribution $\mathcal{N}(\hat{\boldsymbol{\zeta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\zeta}})$ in feature space with parameters

$$\begin{aligned} \hat{\boldsymbol{\zeta}} &= \boldsymbol{S}_{\boldsymbol{\zeta}}\boldsymbol{\zeta}_1 + \boldsymbol{S}_{\boldsymbol{u}}\hat{\boldsymbol{u}}, \\ \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\zeta}} &= \boldsymbol{S}_{\boldsymbol{u}}(\boldsymbol{S}_{\boldsymbol{u}}^\top\boldsymbol{Q}_{\boldsymbol{s}}\boldsymbol{S}_{\boldsymbol{u}} + \boldsymbol{R}_{\boldsymbol{s}})^{-1}\boldsymbol{S}_{\boldsymbol{u}}^\top. \end{aligned}$$

With the above interpretation, we can see that the most basic form of MPC provides a convenient structure to synthesize movements from a probabilistic generative model. We will illustrate this with the example of a *Gaussian mixture model* (GMM) to fit N samples $\{\boldsymbol{\zeta_t}\}_{t=1}^{N}$ with K Gaussians of parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, with centers $\boldsymbol{\mu}_k$ and full covariance matrices $\boldsymbol{\Sigma}_k$. We define a sequence of discrete latent variables $\boldsymbol{s}$ indicating for each time step the Gaussian component corresponding to the motion state (e.g., $\boldsymbol{s} = \{1, 1, 2, 2, 2, 3, \ldots, K\}$). The evolution of a complete movement $\boldsymbol{\zeta}$ can then be represented with a stepwise reference defined as $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{s_t}$ and a full precision matrix defined by $\boldsymbol{Q}_t = \boldsymbol{\Sigma}_{s_t}^{-1}$.

By providing a control cost (e.g., with $\boldsymbol{R}_t = \rho \boldsymbol{I}$ and ρ set in regard to the desired smoothness), the above linear quadratic tracking problem can be solved in closed form, and provides a sequence of commands that can be used for further processing or to reconstruct a trajectory. This approach has links with trajectory models used in the field of speech processing [3], but the MPC approach provides a more powerful formulation with a direct link to the generative process (it can retrieve a controller instead of a trajectory).

We can thus easily associate a probabilistic representation such as GMM with a cost function typical to MPC, where the methods developed for MPC can be used for both control and motion synthesis. The probabilistic representation provides a way to automatically determine the tracking weights and references in the cost function, notably in the form of full precision matrices $\boldsymbol{Q}_t$, which would be too difficult to design manually. It also offers a Bayesian perspective to MPC, by enabling the resulting controller to be viewed as a distribution over control commands (or over a trajectory after reconstruction), whose error terms propagate from the compact mixture model to the resulting controller.

When combined with statistical learning techniques, the most basic form of MPC, as in the above, already shows great promises for planning and control. Several extensions can further be considered, such as extending the problem to inequality constraints, extending the distribution to more powerful generative models such as hidden Markov models, Gaussian processes or other forms of trajectory distributions [4], extending the representation to subspace clustering [1], or sharing components through the use of semi-tied covariances [5].

III. Extension to multiple coordinate systems

We discussed in the above the exploitation of variability and coordination information in MPC, but we did not discuss in which coordinate system the analysis and retrieval should take place.

Skillful movement planning and control require the orchestration of multiple coordinate systems that can have varying levels of importance along the task [6]. Typical examples are movements in object-, body-, head-, or gaze-centered frames of reference that can collaborate in various manners for the different phases of a task. Invariance and coordination extraction in movements are also closely linked to the coordinate systems in which the analysis takes place [7].

Our work takes inspiration from these lines of work to build an extension of GMM to encode movements from the perspective of multiple frames of reference, where the statistical analysis is simultaneously conducted in each coordinate system by replacing the standard GMM with a tensor-variate version of GMM. Such model corresponds to a set of Gaussians encoding the variations and coordinations in each frame, see [1] for details. With this representation, the orchestration and transition between a set of P coordinate systems can be learned from data samples and applied to new situations by extending the cost function (4) to

$$c = \sum_{j=1}^{P} \big(\boldsymbol{\mu}_{\boldsymbol{s}}^{(j)} - \boldsymbol{\zeta}\big)^{\!\top} \boldsymbol{Q}_{\boldsymbol{s}}^{(j)} \big(\boldsymbol{\mu}_{\boldsymbol{s}}^{(j)} - \boldsymbol{\zeta}\big) \;+\; \boldsymbol{u}^{\!\top} \boldsymbol{R}_{\boldsymbol{s}} \boldsymbol{u},$$

with control commands then becoming

$$\mathcal{N}\Big(\hat{\boldsymbol{u}},\, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}}\Big) \;\sim\; \mathcal{N}\Big(\boldsymbol{0},\, \boldsymbol{R}_{\boldsymbol{s}}^{-1}\Big) \prod_{j=1}^{P} \mathcal{N}\Big(\boldsymbol{\mu}_{\boldsymbol{u}}^{(j)},\, {\boldsymbol{\Gamma}_{\boldsymbol{u}}^{(j)}}^{-1}\Big),$$

$$\begin{aligned}
\text{with} \quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} &= \left(\sum_{j=1}^{P} \boldsymbol{\Gamma}_{\boldsymbol{u}}^{(j)} + \boldsymbol{R}_{\boldsymbol{s}}\right)^{-1} \\
&= \left(\sum_{j=1}^{P} \boldsymbol{S}_{\boldsymbol{u}}^{\!\top} \boldsymbol{Q}_{\boldsymbol{s}}^{(j)} \boldsymbol{S}_{\boldsymbol{u}} + \boldsymbol{R}_{\boldsymbol{s}}\right)^{-1}, \\
\hat{\boldsymbol{u}} &= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} \sum_{j=1}^{P} \boldsymbol{\Gamma}_{\boldsymbol{u}}^{(j)} \boldsymbol{\mu}_{\boldsymbol{u}}^{(j)} \\
&= \hat{\boldsymbol{\Sigma}}_{\boldsymbol{u}} \sum_{j=1}^{P} \boldsymbol{S}_{\boldsymbol{u}}^{\!\top} \boldsymbol{Q}_{\boldsymbol{s}}^{(j)} (\boldsymbol{\mu}_{\boldsymbol{s}}^{(j)} - \boldsymbol{S}_{\boldsymbol{\zeta}} \boldsymbol{\zeta}_1).
\end{aligned}$$

where $\boldsymbol{\mu}_{\boldsymbol{u}}^{(j)} = \tilde{\boldsymbol{S}}_{\boldsymbol{u}} (\boldsymbol{\mu}_{\boldsymbol{s}}^{(j)} - \boldsymbol{S}_{\boldsymbol{\zeta}} \boldsymbol{\zeta}_1)$ and $\boldsymbol{\Gamma}_{\boldsymbol{u}}^{(j)} = \boldsymbol{S}_{\boldsymbol{u}}^{\!\top} \boldsymbol{Q}_{\boldsymbol{s}}^{(j)} \boldsymbol{S}_{\boldsymbol{u}}$, thus extending (5) to the consideration of targets in multiple coordinate systems.

IV. Applications and future work

The approach of considering multiple frames of reference to extract relevant features and re-use those in new situations can be exploited in various learning and control applications.

Thus far, it was exploited in [4] for adaptation to new configurations of the environment by considering coordinate systems representing objects or virtual landmarks in the environment. In this application, a grasping movement is observed from the perspective of two coordinate systems corresponding to the robot and to the object to grasp. During reproduction, the robot generalizes the grasping movement to new situations (new locations of object) by smoothly switching from a natural pose in the robot workspace to an appropriate movement to approach and grasp the object.

A preliminary version of the approach was exploited in [8] for adaptation to the user in collaboration skills. Here, the collaborative tasks of transporting objects and assembling furnitures were considered. In the latter, the robot needs to hold the tabletop stiffly when the operator screws one of the table leg, but needs to remain compliant in other phases of the collaboration, in order to allow the user to reorient and move the table for a comfortable and ergonomic work, which is learned by the robot based on observations of the collaborative task executed by two persons.

The approach was also used in [5] to handle shared autonomy and semi-autonomous behaviors in teleoperation. Here, the challenge is to exploit previous recordings of a teleoperated valve turning operation to adapt the skill to new situations (including different operating ranges), while allowing the teleoperator to take part in the operation by exploiting the multiple frames of reference mechanism as a trade-off between autonomous behaviors (when the task is well known) and assisted teleoperation (when the task has multiple options or variants that should be determined by the user).

Finally, it was exploited in [1] to study task priority learning and retrieval with a bimanual reaching example. Here, the goal is to exploit the frames of reference mechanism to provide a set of candidate nullspace projection operators (which are locally linear). The robot then determines from statistical analysis of the set of observations (projected in different nullspaces) in which manner the priority levels are organized and change during the task. This information can then be used to generalize the task to new situations while maintaining the demonstrated prioritization behavior.

The applications above illustrate the capability of combining probabilistic models with MPC for planning purpose, but many other perspectives can stem from this approach. Promising directions of research include the use of Gaussian conditioning to provide contextual information to the system, so that other modality such as force information can be used as additional inputs to drive the system. The probabilistic form of the underlying representation and the retrieved controller readily enable such computation.

An extension to subspace clustering can also be employed to handle sensory data of potentially high dimension. Models such as mixture of factor analyzers (MFA) appear as interesting candidates for reduced motion models preserving the property described above. In the context of MPC, these techniques could for example be used to encapsulate the most salient correlational information and to simplify the search by guiding the exploration toward preferred coordination patterns.

Another important property offered by the probabilistic treatment of MPC is that the resulting distributions can be used for stochastic sampling of new movements. The generated samples will emulate the characteristics of the original samples used to train the model. This approach provides a way to search for new solutions, in a boundary that remains adapted to the initial set of demonstrations.

As a more general remark, combining learning, planning and control in robotics often has the pernicious effect of passing information from one subproblem to the other by keeping only an average and, thus, by discarding higher order statistics. This is typically the case when a single trajectory is used as a reference to be tracked, while, as discussed in the above, most skills are better described by coordination patterns or by variations allowed by the task. For some skills, one can argue that the covariance is more informative than the mean of the distribution, which is the case when a coordination pattern needs to be maintained or when some directions are not relevant for the task (e.g., holding a filled glass requires two out of three Euler angles to be tracked to remain horizontal).

Keeping a distribution of control commands with an underlying sparse representation thus appears as advantageous, and is particularly relevant when combining MPC with other processes. It is in particular important for applications in which the user is in close proximity to the robot (including collaborative robots, assistive robots or prosthetics), where the robot needs to quickly adapt to changes in the environment and to remain compliant when it is appropriate.

APPENDIX I
MULTIVARIATE NORMAL DISTRIBUTIONS

The product of two Gaussians $\mathcal{N}(\boldsymbol{\mu}^{(1)},\boldsymbol{\Sigma}^{(1)})$ and $\mathcal{N}(\boldsymbol{\mu}^{(2)},\boldsymbol{\Sigma}^{(2)})$ is defined by

$$\begin{aligned}\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) &\sim \mathcal{N}(\boldsymbol{\mu}^{(1)},\boldsymbol{\Sigma}^{(1)})\,\mathcal{N}(\boldsymbol{\mu}^{(2)},\boldsymbol{\Sigma}^{(2)}),\\ \text{with}\qquad \boldsymbol{\Sigma} &= \Big({\boldsymbol{\Sigma}^{(1)}}^{-1}+{\boldsymbol{\Sigma}^{(2)}}^{-1}\Big)^{-1},\\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}\Big({\boldsymbol{\Sigma}^{(1)}}^{-1}\boldsymbol{\mu}^{(1)}+{\boldsymbol{\Sigma}^{(2)}}^{-1}\boldsymbol{\mu}^{(2)}\Big).\end{aligned}$$

In the above, $\boldsymbol{\mu}$ minimizes the quadratic cost

$$\begin{aligned}c &= (\boldsymbol{x}-\boldsymbol{\mu}^{(1)})^{\top}{\boldsymbol{\Sigma}^{(1)}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})\\ &\quad+(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})^{\top}{\boldsymbol{\Sigma}^{(2)}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)}),\end{aligned}$$

with $\boldsymbol{\Sigma}$ the error on the estimate.

Another important property of Gaussians is that if $\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, the linear transformation $\boldsymbol{A}\boldsymbol{x}+\boldsymbol{c}$ follows the distribution

$$\boldsymbol{A}\boldsymbol{x}+\boldsymbol{c}\sim\mathcal{N}(\boldsymbol{A}\boldsymbol{\mu}+\boldsymbol{c},\ \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^{\top}).$$

APPENDIX II
TRANSFER MATRICES IN LIFTED VECTOR FORM

The transfer matrices can be retrieved by expressing all future states $\boldsymbol{\zeta}_t$ as an explicit function of the state $\boldsymbol{\zeta}_1$. In the special case of $\boldsymbol{A}_t=\boldsymbol{A}$ and $\boldsymbol{B}_t=\boldsymbol{B},\forall t\in\{1,\ldots,T\}$, which is for example the case when considering a canonical double integrator system, we can write

$$\begin{aligned}\boldsymbol{\zeta}_2 &= \boldsymbol{A}\boldsymbol{\zeta}_1+\boldsymbol{B}\boldsymbol{u}_1,\\ \boldsymbol{\zeta}_3 &= \boldsymbol{A}\boldsymbol{\zeta}_2+\boldsymbol{B}\boldsymbol{u}_2=\boldsymbol{A}(\boldsymbol{A}\boldsymbol{\zeta}_1+\boldsymbol{B}\boldsymbol{u}_1)+\boldsymbol{B}\boldsymbol{u}_2,\\ &\ \vdots\\ \boldsymbol{\zeta}_T &= \boldsymbol{A}^{T-1}\boldsymbol{\zeta}_1+\boldsymbol{A}^{T-2}\boldsymbol{B}\boldsymbol{u}_1+\boldsymbol{A}^{T-3}\boldsymbol{B}\boldsymbol{u}_2+\cdots+\boldsymbol{B}\boldsymbol{u}_{T-1},\end{aligned}$$

which can be organized in a matrix form as

$$\underbrace{\begin{bmatrix}\boldsymbol{\zeta}_1\\ \boldsymbol{\zeta}_2\\ \boldsymbol{\zeta}_3\\ \vdots\\ \boldsymbol{\zeta}_T\end{bmatrix}}_{\boldsymbol{\zeta}}=\underbrace{\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{A}\\ \boldsymbol{A}^2\\ \vdots\\ \boldsymbol{A}^{T-1}\end{bmatrix}}_{\boldsymbol{S}_{\boldsymbol{\zeta}}}\boldsymbol{\zeta}_1+\underbrace{\begin{bmatrix}\boldsymbol{0}&\boldsymbol{0}&\cdots&\boldsymbol{0}&\boldsymbol{0}\\ \boldsymbol{B}&\boldsymbol{0}&\cdots&\boldsymbol{0}&\boldsymbol{0}\\ \boldsymbol{A}\boldsymbol{B}&\boldsymbol{B}&\cdots&\boldsymbol{0}&\boldsymbol{0}\\ \vdots&\vdots&\ddots&\vdots&\vdots\\ \boldsymbol{A}^{T-2}\boldsymbol{B}&\boldsymbol{A}^{T-3}\boldsymbol{B}&\cdots&\boldsymbol{B}&\boldsymbol{0}\end{bmatrix}}_{\boldsymbol{S}_{\boldsymbol{u}}}\underbrace{\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\\ \vdots\\ \boldsymbol{u}_T\end{bmatrix}}_{\boldsymbol{u}}.$$

Similar transfer matrices can be computed for a time horizon T in which $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ are local linearization of the system.

REFERENCES

[1] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, January 2016.

[2] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature Neuroscience*, vol. 5, pp. 1226–1235, 2002.

[3] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech and Language*, vol. 21, no. 1, pp. 153–173, 2007.

[4] M. J. A. Zeestraten, S. Calinon, and D. G. Caldwell, "Variable duration movement encoding with minimal intervention control," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016, pp. 497–503.

[5] A. K. Tanwani and S. Calinon, "Learning robot manipulation tasks with task-parameterized semi-tied hidden semi-Markov model," *IEEE Robotics and Automation Letters (RA-L)*, vol. 1, no. 1, pp. 235–242, January 2016.

[6] D. Bennequin, R. Fuchs, A. Berthoz, and T. Flash, "Movement timing and invariance arise from several geometries," *PLoS Comput. Biol.*, vol. 5, no. 7, pp. 1–27, 2009.

[7] D. Sternad, S.-W. Park, H. Mueller, and N. Hogan, "Coordinate dependence of variability analysis," *PLoS Comput. Biol.*, vol. 6, no. 4, pp. 1–16, 04 2010.

[8] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Trans. on Robotics*, vol. 32, no. 3, pp. 513–527, June 2016.