Dynamical Systems in Robotics: Learning for Control

Exploring Nonlinear Dynamics in Robotic Operations

Alessio De Bona

Industrial Robotics - DIISM

Prof. Gionata Salvietti

March 4, 2024

Overview

The aim is a versatile framework designed for acquiring the non-linear dynamics of motion in manipulation tasks and generating dynamic laws for overseeing both position and orientation control.

This research is based on *Learning from Demonstration*, wherein the dynamics of arm motion are processed as multivariate dynamical systems.

Overview

The efficacy of this approach can be categorized into three aspects:

1) It derives dynamical control laws from demonstrations, facilitating concurrent and seamless control of position and orientation.

In addressing the escalating demand for intuitive control methods, LfD has emerged as a viable approach. LfD establishes user-friendly techniques, allowing a human user to visually instruct a robot on performing a particular task.

Overview

2) It enables the generalization of a motion from unexplored areas of the operational space.

The immediate outcome of applying the functions to a new set of input variables is the automatic extension or generalization of motions to previously unobserved regions of space.

Overview

3) It ensures real-time motion adaptation in the presence of spatial and temporal disturbances.

Under perturbations, a generic motion has to be re-planned, and this process may be too slow to be computed in real-time if perturbations are frequent. This new system is instead capable of promptly recovering from a spatiotemporal perturbation in real-time guaranteeing the convergence to the objective pose

Approach

Implementing a Gaussian Mixture Model allows to directly encapsulate the multivariate dynamics of a motion.

This approach offers several advantages:

- it enables simultaneous control of both the translational and orientation aspects of a motion;
- the system is autonomous, meaning it is not dependent on time, thereby enhancing resilience to perturbations that may deviate the system from its intended trajectory;
- the encoding incorporates the correlation among the dynamics influencing each variable, ensuring the preservation of correlation across variables and the motion dynamics of each variable individually.

Learning a Control Law

We aim to acquire a control law for the generation of robot motion through the utilization of time-invariant DS. The robotic system's motion can be entirely defined by its state, denoted as $X \in \mathbb{R}^N$, and is delineated by a set of ordinary differential equations (ODEs).

We postulate the existence of a first-order, autonomous DS, represented by $f(x)$, which serves as a descriptor for a nominal motion plan for the robot.

$$f : \mathbb{R}^N \to \mathbb{R}^N$$

$$\dot{x} = f(x)$$

where $f : \mathbb{R}^N \to \mathbb{R}^N$ is a continuously differentiable, vector-valued function representing the dynamics of the system. Learning consists of estimating the function f, which maps the N-dimensional input state $x \in \mathbb{R}^N$ to its time-derivative $\dot{x}$.

Problem Statement

The learning algorithm aims to precisely replicate training data within the DS while also extending its capabilities to generalize over unexplored regions. This dual focus ensures **accuracy** in known scenarios and **adaptability** in unfamiliar contexts.

One desirable property is stability at an attractor, x^*. Hence, f must be such that

$$\dot{x}^* = f(x^*) = 0$$

$$\lim_{t \to \infty} x = x^*$$

In this context, $f() : \mathbb{R}^N \to \mathbb{R}^N$ denotes a continuous and differentiable function with a vector output, characterizing a dynamical system (DS). This DS is designed to converge towards a singular stable equilibrium point denoted as x^*, commonly referred to as the **attractor**.

Learning Multivariate Dynamics

In acquiring the function, we opt for an expression parameterized by a set of parameters denoted as θ, shaping the function as $f(x;\theta)$.

The essence of learning lies in the iterative adjustment of these parameters, to progressively align the function with the reference trajectories.

The degree of alignment is quantified using a loss function, represented as $L(X, f, \theta)$, which serves as a measure of how closely the learned function matches the reference trajectories.

Formulation

Given a set of M reference trajectories:

$$\{\mathbf{X}, \dot{\mathbf{X}}\} = \left\{X^m, \dot{X}^m\right\}_{m=1}^{M} = \left\{\left\{x^{t,m}, \dot{x}^{t,m}\right\}_{t=1}^{T_m}\right\}_{m=1}^{M}$$

Being T_m the length of each m^{th} trajectory, the objective is to estimate the parameters Θ of the function $f(x; \Theta)$ to optimally represent the reference trajectories, minimizing the loss $L(X, f, \Theta)$.

Moreover, the function f is expected not only to replicate the dynamics of the reference trajectories accurately but also to exhibit the ability to generate motion with characteristics akin to those trajectories in regions of the state space not covered during training. This aspect is evaluated by quantifying the loss on a testing set comprising reference trajectories not employed in the training phase.

Formulation

Lastly, the function f is required to ensure the convergence of the system to the target x^* from any initial point in space.

In essence, the learning objectives can be succinctly summarized as follows:

1. **Reproduce the reference dynamics**: Optimize the function to closely emulate the dynamics of the reference trajectories.
2. **Converge to the attractor:** Guarantee that the learned function guides the system to converge towards the attractor x^* regardless of the initial conditions.

From a machine learning perspective, estimating $\dot{x} = f(x)$ from data can be framed as a regression problem, where the inputs are the state variables x and the outputs are their first-order derivatives $\dot{x}$.

Unfortunately, standard machine learning techniques can ensure objective (1) but not objective (2).

Gaussian Mixture Regression for DS Learning

With GMR, a first-order DS, $\dot{x} = f(x)$, is estimated by learning about a joint density of position and velocity measurements through a K-component Gaussian mixture model (GMM) as follows:

$$p\left(x, \dot{x} \mid \Theta_{GMR}\right) = \sum_{k=1}^{K} \pi_k N(\cdot \mid \mu^k, \Sigma^k)$$

with $(\cdot \mid \mu, \Sigma)$ being the multivariate Gaussian (or normal) distribution; and π_k represent the priors of each Gaussian component, with $\sum_{k=1}^{K} \pi_k = 1$. Each k^{th} Gaussian distribution is parameterized by

$$\theta_k = \left\{\mu^k, \Sigma^k\right\}, \text{ where } \mu^k = \begin{bmatrix} \mu_x^k \\ \mu_{\dot{x}}^k \end{bmatrix} \text{ and } \Sigma^k = \begin{bmatrix} \Sigma_x^k & \Sigma_{x\dot{x}}^k \\ \Sigma_{\dot{x}x}^k & \Sigma_{\dot{x}}^k \end{bmatrix}$$

Gaussian Mixture Regression for DS Learning

The parameters $\Theta_{GMR} = \{\pi_k, \theta_k\}_{k=1}^{K}$ can be estimated through Maximum likelihood (ML) estimation via an iterative expectation maximization (EM) algorithm. The number of components K can be selected via either *model selection* or *cross-validation* approaches.

For the former, model selection metrics are used to find the optimal number of parameters that trade off the likelihood of the model with the model complexity. For the latter, regression metrics, such as mean squared error (MSE) variants, are used to minimize the error between the estimated velocities and the observed ones; that is, minimize

$$\frac{1}{L}\sum_{i=1}^{L} \left\| f\left(x_i\right) - \dot{x}_i \right\| \text{ for } L \text{ data points.}$$

Learning Multivariate Dynamics

Once Θ_{GMR} and K are inferred, the learned DS is obtained by computing the expectation over the conditional density $\mathbb{E}\{p(\dot{x} \mid x)\}$ as follows:

$$\dot{x} = f\left(x; \Theta_{\mathrm{GMR}}\right) = \mathbb{E}\{p(\dot{x} \mid x)\} = \sum_{k=1}^{K} \gamma_k(x) \tilde{\mu}^k(x),$$

with

$$\gamma_k(x) = \frac{\pi_k p\left(x \mid \mu_x^k, \Sigma_x^k\right)}{\sum_{i=1}^{K} \pi_i p\left(x \mid \mu_x^i, \Sigma_x^i\right)}, \quad \tilde{\mu}^k(x) = \mu_{\dot{x}}^k + \Sigma_{\dot{x}x}^k \left(\Sigma_x^k\right)^{-1} \left(x - \mu_x^k\right).$$

The terms $\tilde{\mu}^k(x)$ can be interpreted as local linear regressive functions whose slope is determined by Σ_{xx}^k (the variance of x) and $\Sigma_{\dot{x}x}^k$ (the covariance of $\dot{x}$ and x).

Learning Multivariate Dynamics

Further, $y_k(x) = p\left(k \mid x, \Theta_{GMR}\right)$ is the a posteriori probability for a data point x belonging to the k^{th} Gaussian component, where $p\left(x \mid \mu_*^*, \Sigma_*^*\right) = N\left(x \mid \mu_*^*, \Sigma_*^*\right)$ is the Gaussian density function equation.

We refer to $\gamma_k(x)$ as the mixing weights for the linear regressors. Note that the mixing functions, $\gamma_k(x)$, hold the following properties:

$$\begin{cases} 0 < \gamma_k(x) \leq 1 \quad \forall k = 1, \ldots, K \\ \sum_{k=1}^{K} \gamma_k(x) = 1. \end{cases}$$

The learned DS, $\dot{x} = f(x)$, is thus a weighted combination of K linear regressive models.

Illustrative 2D reference trajectories for DS learning with standard regression algorithms

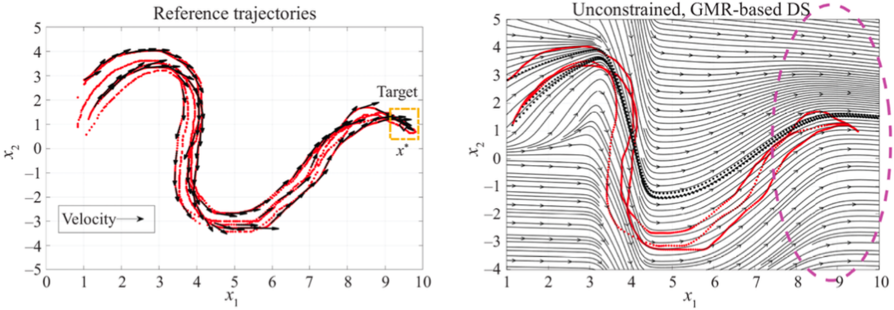


Figure: **Red** trajectories are demonstrations, and **black** trajectories are reproductions of the learned DS from the same initial state as demonstrations

Unstable DS regressors

As depicted in the preceding image, utilizing standard regression algorithms like GMR leads to an unstable Dynamical System (DS). This instability is evident in two aspects:

1. The DS tends to diverge in regions of the state space where no data points were collected.
2. There is no assurance that the motion generated by the DS will come to a halt at the intended target.

This instability is a result of the regressive function parameters (i.e., Θ_{GMR}) not being constrained to ensure the avoidance of the aforementioned issues.

Unstable DS regressors

Such instabilities arise as the learned regressive function parameters (i.e., θ_{GMR}), are not constrained to guarantee the absence of 1 and 2.

In other words, they are not forced to ensure the global asymptotic stability (GAS) constraints expressed in the equation $\dot{\bar{x}} = f(\bar{x}) = 0$

To enforce the stability constraints expressed by the equation $\dot{\bar{x}} = f(\bar{x}) = 0$ for a dynamical system (DS) mapping function $f(x; \theta^*) : \mathbb{R}^N \rightarrow \mathbb{R}^N$, conditions must be directly imposed on the learned parameters θ^* derived via Lyapunov's second method for GAS systems.

Lyapunov Theory for Stable DSs

A function $\dot{x} = f(x)$ is a DS that is GAS at the attractor $x^* \in \mathbb{R}^N$, if there exists a continuous and continuously differentiable Lyapunov candidate function $V(x) : \mathbb{R}^N \rightarrow \mathbb{R}$ that is radially unbounded; that is, $\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$ and satisfies the following conditions:

- $V\left(x^*\right) = 0,$
- $V(x) > 0 \forall x \in \mathbb{R}^N \backslash x = x^*$
- $\dot{V}\left(\mathrm{x}^*\right) = 0,$
- $\dot{V}(x) < 0 \forall x \in \mathbb{R}^N \backslash x = x^*$

Intuitively, this theorem states that for any DS of the form given in equation $\dot{\bar{x}} = f(\bar{x}) = 0$ to be GAS there should be a corresponding energy-like function $V(x)$ which should be non-increasing along all trajectories of $f(x)$.

Lyapunov Theory for Stable DSs

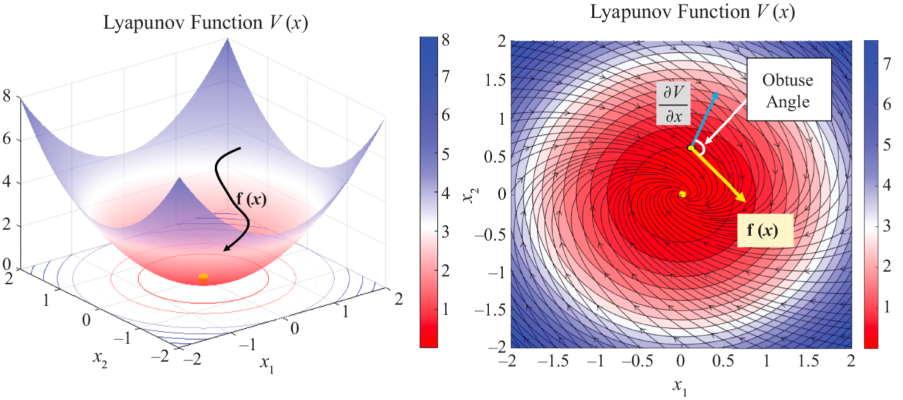


Figure: Left: Illustration of a Lyapunov function. Right: Geometrical intuition of the theorem

Lyapunov Theory for Stable DSs

In other words, if

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x} f(x) < 0,$$

then the angle between the gradient of the Lyapunov function $\nabla V(x) = \frac{\partial V(x)}{\partial x}$ and the direction of motion of the vector field $f(x)$ should be obtuse, as depicted in the right figure.

This guarantees that the trajectories of $f(x)$ are directed toward lower values of the Lyapunov function $V(x)$ at all times. Hence, a GAS DS must have a corresponding positive definite function $V(x)$ ensuring that the conditions are met.

Lyapunov Theory for Stable DSs

In control theory, the most common Lyapunov function used for linear and/or linearized DS is the quadratic Lyapunov function, (QLF) that is,

$$V(x) = \frac{1}{2}\left(x - x^*\right)^T \left(x - x^*\right).$$

Sufficient stability conditions for a linear time-invariant (LTI) DS are of the following form:

$$\dot{x} = f(x) = Ax + b,$$

where $b \in \mathbb{R}^N$ can be seen as an offset or translation of the origin of the DS and $A \in \mathbb{R}^{N\times N}$ is the linear system matrix that defines the dynamics of the LTI system.

Lyapunov Theory for Stable DSs

The linear DS is GAS at the attractor x^* if,

$$\begin{cases} b = -Ax^* \\ A^T + A \prec 0, \end{cases}$$

where $\prec$ refers to the negative definiteness of a matrix, respectively. A matrix A is deemed negative definite if its symmetric part $\tilde{\mathrm{A}} = \frac{1}{2}\left(\ \mathrm{A}^{\mathrm{T}} + \mathrm{A}\right)$ has all negative eigenvalues.

Lyapunov Theory for Stable DSs

Often, however, using a basic QLF to ensure the stability of a DS might result in overly conservative constraints that restrict the complexity of the dynamics of $f(x)$. To alleviate this, another common Lyapunov function can be used, the parameterized quadratic Lyapunov function (P-QLF), which has the following form:

$$V(x) = (x - x^*)^T P(x - x^*)$$

where $P \in \mathbb{S}_N^+ \subset \mathbb{R}^{N \times N}$ is a symmetric positive definite matrix that reshapes a QLF. Such reshaping offers a less strict stability condition that permits stronger non-linearities in the dynamics of the corresponding DS $f(x)$.

Nonlinear DSs as a Mixture of Linear Systems

The Lyapunov stability conditions derived from a QLF or P-QLF are applicable exclusively to *linear dynamical systems* represented by $\dot{x} = f(x) = Ax + b$.

To address nonlinear systems, a formulation involving a mixture of linear systems is employed:

$$\dot{x} = f(x; \theta_{GMR}) = E\{p(\dot{x}|x)\} = \sum_{k=1}^{K} \gamma_k(x)(A^k x + b^k)$$

$A^k \in \mathbb{N} \times \mathbb{N}$ and $b^k \in \mathbb{R}^N$ represent the parameters of the k^{th} linear system. The function $\gamma_k(x) : \mathbb{R}^N \rightarrow \mathbb{R}^+$, is known as mixing. The summation over k allows for the representation of the nonlinear dynamics as a combination of weighted linear systems, providing a more versatile framework for analysis and control than traditional linear approaches.

Learning Stable Nonlinear DSs

The formulation of the nonlinear dynamical system (DS) as a mixture of linear DS draws parallels with the structure of a Gaussian Mixture Regressor (GMR).

Specifically, just as the nonlinear DS is a blend of linear DS, the GMR is a combination of linear regressors. A constrained GMR learning algorithm is now introduced and named the Stable Estimator of DSs (SEDS).

Constrained Gaussian Mixture Regression

Let's begin by defining the nonlinear DS parameters of the equation (i.e., $\Theta_{f(x)} = \{\gamma_k(x), A^k, b^k\}_{k=1}^{K}$), via GMR.

Next, we restate and expand the GMR regressor equation that is, the posterior mean of the conditional distribution $p(\dot{x} \mid x)$ as follows:

$$\dot{x} = f\left(x; \Theta_{GMR}\right) = E\{p(\dot{x} \mid x)\} =$$
$$\sum_{k=1}^{K} \frac{\pi_k p\left(x \mid \mu_x^k, \Sigma_x^k\right)}{\sum_{i=1}^{K} \pi_i p\left(x \mid \mu_x^i, \Sigma_x^i\right)} \left(\mu_{\dot{x}}^k + \Sigma_{\dot{x}x}^k \left(\Sigma_x^k\right)^{-1} \left(x - \mu_x^k\right)\right)$$

Constrained Gaussian Mixture Regression

This expanded GMR regressor equation can be simplified through a change of variables. Let us define this as follows:

$$\begin{cases} A^k = \Sigma^k_{\dot{x}x}\left(\Sigma^k_x\right)^{-1} \\ b^k = \mu^k_{\dot{x}} - A^k \mu^k_x \\ \gamma_k(x) = \frac{\pi_k p\left(x|\mu^k_x,\Sigma^k_x\right)}{\sum_{i=1}^K \pi_i p(x|\mu^i_x,\Sigma^i_x)} \end{cases}$$

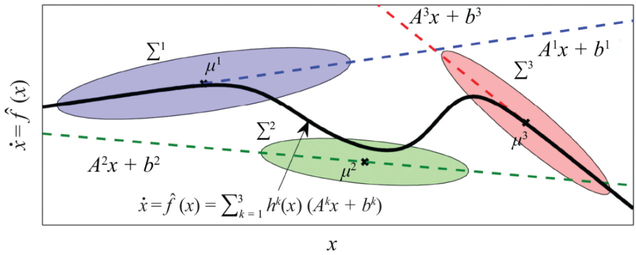
And obtain:

$$\dot{x} = f\left(x;\Theta_{\mathrm{GMR}}\right) = \mathbb{E}\{p(\dot{x} \mid x)\} = \sum_{k=1}^{K} \gamma_k(x)\left(A^k x + b^k\right).$$

Constrained Gaussian Mixture Regression

As can be seen, it is identical to the mixture of linear DS formulations. Using such reparametrization, it can be analyzed the influence of GMR parameters, $\Theta_{GMR} = \{\pi_k, \mu^k, \Sigma^k\}_{k=1}^{K}$, on the resulting DS.

Each linear dynamics $A^k x + b^k$ corresponds to a line that passes through the centers of the Gaussian μ^k with slope A^k that is determined by Σ_{xx}^k (the variance of x) and $\Sigma_{\dot{x}x}^k$ (the covariance of x and $\dot{x}$).



In other words, they define the regions in the state space in which a K^{th} linear DS has more weight or importance.

GAS stability for non-linear case

Previously, the GMR regressor equation underwent reparametrization into a specific form, without guaranteeing GAS at a target x^*. To ensure such a convergence property, a set of sufficient conditions is proposed, derived from a QLF equation.

The nonlinear DS, parametrized by $\Theta_{\mathrm{GMR}} = \left\{\pi_{\mathrm{k}}, \mu^{\mathrm{k}}, \Sigma^{\mathrm{k}}\right\}_{\mathrm{k}=1}^{\mathrm{K}}$, is GAS at the target $x^* \in \mathbb{R}^N$ if

$$\begin{cases} b^k = -A^k x^* \\ A^k + \left(A^k\right)^T \prec 0 \end{cases} \quad \forall k = 1 \ldots K,$$

where $\left(\mathrm{A}^{\mathrm{k}}\right)^{\mathrm{T}}$ is the transpose of A^{k}, and $\prec 0$ refers to the negative definiteness of *a* matrix.

Stable Estimator of DSs

Now, compute the unknown parameters $\Theta_{\mathrm{GMR}} = \{\pi_{\mathrm{k}}, \mu^{\mathrm{k}}, \Sigma^{\mathrm{k}}\}_{\mathrm{k}=\mathbf{1}}^{\mathrm{K}}$ for the Stable Estimator of Dynamical Systems (SEDS). SEDS is a learning algorithm that determines the optimal values of Θ_{GMR} by solving an optimization problem under GAS constraints.

The parameters of a GMR regressor, Θ_{GMR}, are typically estimated by maximizing the likelihood or posterior probability of the Gaussian Mixture Model (GMM) representing the joint distribution of inputs/outputs, $p(\dot{x}, x)$.

In essence, this approach aims to estimate the parameters of the joint probability distribution that best characterizes the observed data.

We, thus, consider two candidates for the objective function of the SEDS algorithm: **log-likelihood**, and **MSE**.

SEDS - [likelihood]

The SEDS [Likelihood] objective function adopts the ML parameter estimation approach for GMM. Similar to the conventional GMM estimation, the parameters of a GMM in SEDS are determined through an Expectation-Maximization (EM) algorithm. The EM algorithm iteratively maximizes the log-likelihood of the GMM.

In this SEDS variant, we maximize the log-likelihood of reference trajectories $\{\mathbf{X}, \dot{\mathbf{X}}\} = \left\{X^m, \dot{X}^m\right\}_{m=1}^{M} = \left\{\{x^{t,m}, \dot{x}^{t,m}\}_{t=1}^{T_m}\right\}_{m=1}^{M}$ described by a GMM with parameters Θ_{GMR}. This optimization is subject to stability constraints and is expressed as follows:

$$\min_{\Theta_{\mathrm{GMR}}} J\left(\Theta_{\mathrm{GMR}}\right) = -\frac{1}{L}\sum_{m=1}^{M}\sum_{t=0}^{T_m}\log p\left(x^{t,m}, \dot{x}^{t,m} \mid \Theta_{\mathrm{GMR}}\right)$$

SEDS - [likelihood]

Subject to:

Stability Conditions

- (a) $b^k = -A^k x^*$
- (b) $A^k + \left(A^k\right)^T \prec 0$

GMM Inherent Constraints

- (c) $\sum^k \succ 0 \quad \forall k = 1, \dots, K$
- (d) $0 < \pi_k \leq 1$
- (e) $\sum_{k=1}^K \pi_k = 1$

where $p\left(x^{t,m}, \dot{x}^{t,m} \mid \Theta_{\mathrm{GMR}}\right)$ is the PDF of the GMM and $L = \sum_{m=1}^M T_m$ is the total number of training data points from the reference trajectories $\{\mathbf{X}, \dot{\mathbf{X}}\}$.Consequently, the mixing/activation function $\gamma(x)$ produces a weighted sum of linear DS.

SEDS - [MSE]

The SEDS Likelihood variant serves as an alternative to the conventional EM algorithm, albeit with specific constraints. In the presence of other regression algorithms such as GPR/SVR is best suited to an alternative objective function based on Mean Squared Error (MSE).

The optimization task involves minimizing the proposed equation, subject to constraints denoted by variables a, b, c, d, and e. The goal is to find the optimal parameters Θ_{GMR} that minimize the defined objective function:

$$\min_{\Theta_{\mathrm{GMR}}} J\left(\Theta_{\mathrm{GMR}}\right) = \frac{1}{2L} \sum_{m=1}^{M} \sum_{t=0}^{T_m} \left\| f\left(x^{t,m}\right) - \dot{x}^{t,m} \right\|^2 .$$

Note: Both the Likelihood and MSE formulations can be cast as nonlinear programming (NLP) problems. Consequently, standard-constrained optimization techniques can be employed to solve these formulations.

Hyper-parameters and Prespecifications

It's important to note that the SEDS initialization algorithm, and consequently, the SEDS learning algorithm, necessitate specifying the number of Gaussian functions K and the attractor of the DS $x^* \in \mathbb{R}^N$.

While the user defines the latter, determining the optimal number of K Gaussians involves utilizing the standard GMM model selection scheme. For GMM model selection, the **Bayesian information criterion (BIC)** metric, for example, is employed to balance a model's likelihood with the necessary parameters for encoding the data.

Parameter Initialization

The choice of initialization method for the optimization parameters, denoted as θ_{GMR}, is influenced by three aspects: the nature of the motion, the quantity of Gaussian functions (K), and the initial approximation of the GMM parameters obtained through the conventional EM algorithm.

When the number of Gaussian functions K is appropriately selected and the conventional EM algorithm effectively fits the Gaussians to the reference trajectories, thus securing the model's maximum likelihood (ML), the initialization methods will apply stability constraints on the covariance matrices, causing minimal alterations to the overall parameter set.

Model and computational complexity

SEDS [MSE] is better than SEDS [Likelihood], in that it requires fewer parameters.

On the other hand, SEDS [MSE] has a more complex cost function that requires computing GMR at each iteration over all training data points. As a result, the use of MSE makes the algorithm computationally more expensive and causes a slightly longer training time.

Developing Robust, Highly Nonlinear Dynamical Systems

An alternative methodology, addressing the fundamental trade-off between precision and stability aims to facilitate progressive learning and reduce dependence on hyper-parameter selection and initial conditions for accuracy:

1. A P-QLF Lyapunov candidate function to ensure stability constraints are met while also capturing nonlinear dynamics more effectively.
2. Maintaining the local behavior of Gaussian functions to support incremental learning processes.
3. Automatically determining the optimal count of K Gaussian functions.

These objectives are realized through the implementation of a Bayesian nonparametric GMM estimation.

Decoupled Linear Parameter Varying Representation

Consider the linear dynamical system (DS) mixture:

$$\dot{x} = f(x) = \sum_{k=1}^{K} \gamma_k(x)(A^k x + b^k)$$

This represents a linear parameter varying (LPV) system, with each $A^k x$ signifying an individual linear time-invariant (LTI) system and $\gamma_k(x)$ constituting a state-dependent parameter vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_K]$.

In LTI systems, Lyapunov functions are commonly of the form:

$$V(x) = (x - x^*)^T P(x - x^*)$$

where stability is assured by the matrix P. Using a P-QLF ensures sufficient conditions to confirm GAS at x^*.

Learning Stable, Highly Nonlinear DSs

A nonlinear DS is GAS at x^* if there exists a symmetric positive definite matrix P, and a Lyapunov function $\mathrm{V}(\mathrm{x}) = (\mathrm{x} - \mathrm{x}^*)^{\mathrm{T}}\mathrm{P}(\mathrm{x} - \mathrm{x}^*)$ such that:

$$\begin{cases} (A^k)^{\mathrm{T}}P + PA^k = Q^k, \quad Q^k = (Q^k)^{\mathrm{T}} \prec 0 \quad \forall k = 1, \dots, K \\ b^k = -A^k x^* \end{cases}$$

By adjusting the entries of P, an anisotropic measure is created, allowing the function to stretch or compress along different axes.

This adjustment is particularly beneficial for modeling trajectories with high curvature or varying speeds of convergence towards x^* across different dimensions.

Learning Stable, Highly Nonlinear DSs

P-QLF complicates the optimization, necessitating simultaneous estimation of P and the matrices A^k. This optimization challenge is nonconvex and heavily reliant on a robust initial estimate of P. To mitigate these complexities, the parameters are optimized incrementally, guided by constraints derived from the P-QLF.

In the traditional SEDS framework, the dynamical system parameters $\Theta_{f(x)} = \{\gamma_k(x), A^k, b^k\}_{k=1}^K$ are closely linked to the GMR parameters $\Theta_{\mathrm{GMR}} = \{\pi_k, \mu^k, \Sigma^k\}_{k=1}^K$, derived from fitting the joint distribution of the reference trajectories. This is a consequence of the variable transformation:

$$\gamma_k(x) = \frac{\pi_k p(x \mid \mu_x^k, \Sigma_x^k)}{\sum_{i=1}^K \pi_i p(x \mid \mu_x^i, \Sigma_x^i)},$$

$$A^k = \Sigma_{\dot{x}x}^k (\Sigma_x^k)^{-1}, \quad b^k = \mu_{\dot{x}}^k - A^k \mu_x^k \quad \forall k = 1, \dots, K$$

Learning Stable, Highly Nonlinear DSs

The SEDS algorithm connects the mixing function $\gamma_k(x)$ with linear DS parameters through means and covariance matrices in a GMM framework. This interdependence makes SEDS nonlinear despite a primarily linear stability constraint. Dissociating parameters requires restructuring the formulation.

$$\dot{x} = f(x, \Theta_{\mathrm{LPV}}) = \sum_{k=1}^{K} \gamma_k(x) \left(A^k x + b^k \right),$$

$$\Theta_{\mathrm{LPV}} = \left\{ \Theta_\gamma = \left\{ \left\{ \pi_k, \theta_\gamma^k = (\mu^k, \Sigma^k) \right\}_{k=1}^{K} \right\}, \Theta_f = \left\{ A^k, b^k \right\}_{k=1}^{K} \right\}.$$

Learning Stable, Highly Nonlinear DSs

By decoupling the parameters, we need only to model the GMM distribution for the position variables [i.e., $p(x|\Theta_\gamma) = \sum_{k=1}^{K} \pi_k N(x|\mu^k, \Sigma^k)$], leaving the linear DS parameter set $\Theta_f = \{A^k, b^k\}_{k=1}^{K}$ unrestricted and open for optimization by any chosen method, separate from Θ_γ.

This separation enables the application of standard semi-definite program (SDP) solvers for linear matrix inequality (LMI), like those in the stability criteria, leading to less stringent constraints.

Additionally, this decoupling preserves the geometric interpretation of the GMM, as its parameters are not altered during the optimization phase with constraints.

Learning Stable, Highly Nonlinear DSs

However, it's important to note that while this LPV-DS framework allows for the use of a P-QLF to establish more flexible stability conditions and to estimate Θ_f using SDPs, the precision of the model reproduction significantly depends on the accurate estimation of Θ_γ.

This entails not only determining the optimal number of Gaussian components K that most accurately represents the reference trajectories but also ensuring that these components are properly aligned with the trajectories, so each Gaussian captures a local segment of the state space that adheres to a linear DS.

Physically Consistent Bayesian Nonparametric GMM

For an accurate reproduction with the LPV-DS formulation, not only is the number of optimal K Gaussian functions important, but they must be consistent with the LPV-DS assumptions *(i.e, each Gaussian should represent data points that follow a quasi-linear motion)*. Such an estimate is empirically hard to find.

While Bayesian nonparametric modeling and estimation approaches can allow one to determine automatically an adequate number of K Gaussian functions, they cannot ensure that each Gaussian would cluster data points that represent linear motion.

We refer to this property as physical consistency.

References

A. Billard, S. Calinon, R. Dillmann, and S. Schaal (2008)
Robot Programming by Demonstration. In the Handbook of Robotics,
MIT Press.

A. Billard, S. Mirrazavi, N. Figueroa (2022)
Learning for Adaptive and Reactive Robot Control A Dynamical Systems Approach
The MIT Press Cambridge, Massachusetts London, England.

S. SunHaihui, G. Li, N. Figueroa (2023)
DAMM: Directionality-Aware Mixture Model Parallel Sampling for Efficient Dynamical System Learning
CC BY 4.0,

E. Gribovskaya A. Billard, (2009)
Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators,
9th IEEE-RAS International Conference on Humanoid Robots, Paris, France

The End