

Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control

S. Mohammad Khansari-Zadeh Klas Kronander Aude Billard
LASA Laboratory, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
Email: {mohammad.khansari, klas.kronander, aude.billard}@epfl.ch

Abstract—Successful execution of many robotic tasks requires precise control of robot motion and its interaction with the environment. In robotics these two problems are mainly studied separately in the domain of robot motion generation and interaction control, respectively. Existing approaches rely on two control loops: a motion generator (planner) that provides a reference trajectory in the outer loop, and an active impedance controller that tracks the reference trajectory in the inner loop. Ensuring stability of the closed-loop system for this control architecture is non-trivial. In this paper, we propose a single-loop control architecture that performs motion generation and interaction control at once. We model robot discrete motions with a time-invariant dynamical system, which is expressed as a nonlinear combination of a set of linear spring-damper systems. This formulation represents the nominal motion and the impedance properties with a single set of parameters, simplifying stability analysis of the closed-loop system. We provide sufficient conditions to ensure global asymptotic stability of this system for movements in free-space, and its passivity during persistent contact with a passive environment. We validate our approach in simulation using the 7-DoF KUKA LWR-IV robot.

I. INTRODUCTION

Motion generation and interaction control are two skills that are jointly essential to safely execute a wide variety of tasks. Robot motion generation focuses on the problem of finding a path from an initial state to a final state given a complete description of the robot’s geometry and its environment [1], whereas robot interaction control is concerned with the problem of describing the robot behavior when it gets into contact with the environment [2]. The application of a pure motion generation approach is limited to free-space movements in well-defined environments (e.g. factories, laboratories), where exact localization of objects is possible [3, 4]. For tasks that require interaction with the environment, these approaches are prone to instability [5, 6, 3, 7].

To overcome this limitation, several approaches endow motion generation with an interaction controller in a two-loop control architecture: a motion generator in the outer loop that provides a reference trajectory, and an impedance controller in the inner loop that tracks the reference trajectory while responding compliantly to external forces arising in contact [8, 9, 10, 11, 12]. The reference trajectory is only followed in the absence

of an external force, and the impedance parameters determine the behavior of the robot in contact. This control architecture also allows execution of an additional repertoire of tasks that requires contact control (such as ironing, grinding, peg in hole), which could not be achieved with a pure motion generation method. As we will elaborate in Section I-A, existing works analyze stability of this control architecture in situations where either the reference trajectory is not updated in closed-loop, or the impedance parameters are fixed across the motion. Ensuring stability of the closed-loop system when using feedback motion planning with a variable impedance control law remains however non-trivial, and to best of our knowledge, no rigorous stability analysis has been done for these systems. In this paper we address this problem and propose a single-loop control architecture that performs feedback motion generation and state varying impedance control at once with guaranteed stability.

A. Problem Statement

Impedance control [2] is one of the prominent interaction control approaches, where the relationship between the manipulator position and contact force is related through tunable impedance parameters. Impedance control can be achieved from two directions: passively through hardware design or actively through the controller. In this paper, we focus on the problem of active impedance control. The classical way to execute a task through motion planning and impedance control is to first plan a trajectory (offline) and then use an impedance controller to follow the generated path during the robot execution.

For a manipulator with d generalized degrees of freedom $\mathbf{s} \in \mathbb{R}^d$, the robot dynamics (whether in operational or joint space) can be represented as [13]:

$$\mathbf{M}(\mathbf{s})\ddot{\mathbf{s}} + \mathbf{C}(\mathbf{s}, \dot{\mathbf{s}})\dot{\mathbf{s}} + \mathbf{g}(\mathbf{s}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad (1)$$

where $\mathbf{M}(\mathbf{s}) \in \mathbb{R}^{d \times d}$ is the mass matrix, $\mathbf{C}(\mathbf{s}, \dot{\mathbf{s}}) \in \mathbb{R}^{d \times d}$ is the Coriolis/centrifugal matrix, $\mathbf{g}(\mathbf{s})$ is the gravitational force, $\boldsymbol{\tau}$ represents the actuators generalized force, and $\boldsymbol{\tau}_{ext}$ is the external generalized force applied to the robot by the environment. Note that if \mathbf{s} is defined in the operational space, special considerations should be taken into account in singular configurations [14].

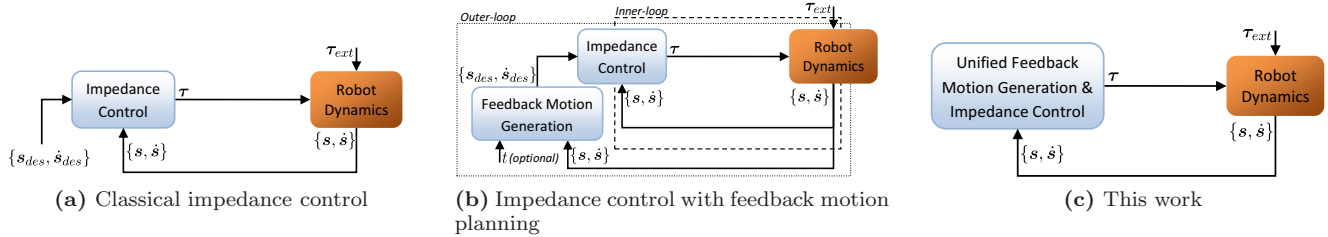


Fig. 1: A simplified schematic comparison between the control architecture of the proposed approach and the two widely used impedance controllers in robotics. By integrating the ‘feedback motion generation’ and ‘impedance control’ blocks into a unified block, the proposed approach facilitates stability analysis of the closed-loop system. For more information refer to [Section I-A](#).

The simplest impedance controller can be designed to mimic the behavior of a spring-damper system [2], i.e.

$$\tau_{IC} = -\mathbf{S}(\mathbf{s} - \mathbf{s}^{des}(t)) - \mathbf{D}(\dot{\mathbf{s}} - \dot{\mathbf{s}}^{des}(t)) \quad (2)$$

where τ_{IC} is the impedance control law¹, $\mathbf{s}^{des}(t)$ and $\dot{\mathbf{s}}^{des}(t)$ are the desired joints position and velocity given by the motion generator. The stiffness \mathbf{S} and damping \mathbf{D} matrices are positive definite and constant. The final actuator command τ that is sent to the robot is given by:

$$\tau = \tau_{IC} + \mathbf{g}(\mathbf{s}) \quad (3)$$

[Figure 1a](#) shows a simple control block diagram of the above approach. In this control architecture the desired trajectory is computed offline and is fed to the controller through time-indexing. Stability of the task is guaranteed as long as \mathbf{S} and \mathbf{D} are positive definite and constant. The impedance control approach with fixed gains is particularly useful in static and controlled environments such as those found in factories. However, since the motion generation does not get feedback from the environment (it runs in open-loop), it is ill-suited for robotic systems that operate in dynamic environments. Furthermore, it also imposes a trade-off between accuracy (using large \mathbf{S} and \mathbf{D}) and safety and energy efficiency (using small \mathbf{S} and \mathbf{D}).

Recent approaches employ a variable impedance control law with the aim to provide higher performance, efficiency, and human safety. In our previous work [15], we introduce a framework that allows the user to interactively shape a task-specific stiffness profile through physical human-robot interaction. This work is then extended in [9] to improve the framework with a modality for locally increasing the stiffness. To avoid closed-loop instability, both [9] and [15] use open-loop reference trajectory in their impedance control law. In [16], a stochastic optimal control is proposed to control variable

¹Throughout this paper, we consider an impedance control law that does not reshape the inherent inertia of the robot. This corresponds to the most common application scenario of impedance control and was hence chosen to maximize the applicability of the paper. However, the results herein extend also to the case where a desired apparent inertia is implemented via force-feedback.

impedance actuators under a predefined tradeoff of task accuracy and energy cost. In [17], a variable impedance control law is introduced to account for dynamics model parameter errors. Reference [18] presents a human-like learning algorithm to model variable impedance control subject to minimization of instability, motion error, and effort. This work is then extended in [19] to perform unstable contact tooling on unknown surfaces.

Stability analysis of variable impedance control is nontrivial and is only evaluated in a few works. In [20], a passivity-based approach is proposed to ensure stability of a time-varying impedance controller. This method evaluates the power balance of the robot and ensures that the amount of energy pumped into the system is always less than the dissipated energy, and thus the system remains passive. In [21], stability analysis of a force tracking impedance controller is presented. References [18] and [19] describe an adaptive approach to guarantee stability of their variable impedance controller. All these approaches analyze stability for cases where the reference trajectory is not updated in closed-loop.

To provide realtime adaptation to changes in dynamic environments, a new body of research is directed at using the combination of variable impedance control with feedback motion planning (see [Fig. 1b](#)). In [8], the authors propose an imitation learning approach to model robot motions with dynamical systems and formulate the stiffness matrix to be inversely proportional to the observed covariance in the demonstrations. A similar formulation is used in [12] with the difference that the impedance parameters are estimated through a reinforcement learning approach (as opposed to learning from demonstrations). References [10, 11] also take a reinforcement learning approach to estimate the variable impedance control law for a given task.

Addition of feedback motion planning to a variable impedance control law aggravates instabilities, and to the best of our knowledge no rigorous stability analysis has been done for these systems. In the works that consider closed-loop update of the reference trajectories [8, 10, 11, 12], motion generation and impedance control are done in two separate loops (see [Fig. 1b](#)). While in

practice, it is often the case that such a system is overall stable, studying separately stability of each control loop is not sufficient to ensure stability of the complete closed-loop system. In other words, the closed-loop system can actually become *unstable* even when both loops are stable separately. In place of theoretical stability analysis, one can proceed to numerical assessment of the stability of the system throughout the state space. However this rapidly becomes computationally intractable as the dimensionality of the system increases.

Despite the importance of ensuring stability of the system (especially when robots must perform tasks in dynamically changing environment or in the vicinity of humans), existing works either do not perform such stability analysis [8, 9] or only evaluate it numerically along a particular desired trajectory using reinforcement learning techniques (i.e. considering local stability) [10, 11, 12].

To summarize, there are two challenges: 1) Providing a rigorous stability analysis for state-varying variable impedance controllers, and 2) Ensuring stability of the system when using feedback motion planners with variable impedance controllers. In this paper we address these problems by proposing a novel approach that unifies ‘motion generation’ and ‘impedance control’ into one single control law. Our *Integrated Motion Generator and Impedance Controller* (i-MOGIC) can be proved to be globally stable without any need to perform numerical stability analysis. Figure 1c illustrates the flow of information in such an integrated control law. To provide realtime adaptability to changing environments and robustness to perturbations, the proposed approach models robot motion as an autonomous Dynamical Systems (DS). This DS is formulated using Gaussian Mixture Regression (GMR) [22], in which each Gaussian function represents a linear spring-damper system. The resulting DS derives from the nonlinear combination of these spring-damper systems, and can be seen as a *state-varying stiffness and damping* system. We provide sufficient conditions to ensure global asymptotic stability of the system, and validate the approach in simulation using the 7-DoF KUKA Light Weight Robot (LWR).

Note that the stability analysis proposed in this paper differs from those in [18, 19, 20, 21] in two important aspects: 1) it is applicable to *state-varying* impedance controller (as opposed to time-varying), and 2) it also includes stability of feedback motion planner. In our previous works [23, 24], we use GMR formulation to model first order autonomous DS. First order DS only represent kinematic information and are hence not well-suited for interaction control. Furthermore, the robot controller in [23, 24] relies on the two-loop control architecture and thus shares the same limitation as described above.

II. DYNAMICAL SYSTEMS-BASED ROBOT MOTIONS

Consider a state variable $\mathbf{s} \in \mathbb{R}^d$ (e.g. \mathbf{s} could be defined in joint or task space). We formulate our impedance control law as a time-invariant DS:

$$\tau_{IC} = \mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) \quad \mathbf{f} : \mathbb{R}^{d \times d} \mapsto \mathbb{R}^d \quad (4)$$

where $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ is a continuous function that codes a specific behavior. Starting in an initial state $(\mathbf{s}^0, \dot{\mathbf{s}}^0)$, the robot actuator command is given by Eqs. (3) and (4), and the robot motion follows according to Eq. (1).

As outlined before, we consider the class of motions which ends to a single point $(\mathbf{s}^*, \dot{\mathbf{s}}^*)$, i.e. the attractor of the system. Typical examples of such motion are swinging a golf club, inserting peg in hole, reaching out for an object, closing fingers in a particular grasping configuration, stepping motion, etc. Note that this does not necessarily imply that the goal of the task is limited to reaching \mathbf{s}^* . For example, in a polishing task, the attractor would simply be the end of one polishing stroke while the ‘goal’ of the task is actually fulfilled during the convergence to the attractor. The function $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ can either be provided by the user, or it can be estimated using some machine learning techniques [25, 26]. In this paper, we focus on developing a suitable mathematical representation for $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ so that it allows combining motion planning and interaction control with guaranteed global asymptotic stability.

III. ROBOT MOTION ENCODING WITH STATE-VARYING STIFFNESS AND DAMPING

There are numerous ways to parameterize a nonlinear DS. In this paper, we formulate our desired control policy $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ using the same structure as Gaussian Mixture Regression (GMR) [22] due to three reasons: 1) As we will show later on in Section III-B, the GMR formalism allows us to describe $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ as a nonlinear weighted sum of spring-damper systems which can be used to shape both the unperturbed motion and reactive behavior in contact, 2) The unique structure of GMR allows us to derive explicit stability conditions on the parameters of Gaussian Mixture Model (GMM) to ensure global asymptotic stability of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ without performing computationally expensive numerical stability analysis, and 3) When it is desired, one could exploit the power of existing machine learning techniques to build an estimate of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$.

A. Gaussian Mixture Regression, a probabilistic view

In this section we provide a brief overview of GMR from a probabilistic perspective. GMR is a nonlinear regression technique that works on the joint probability $\mathcal{P}([\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}])$ between input $\zeta_{\mathcal{I}} \in \mathbb{R}^n$ and output $\zeta_{\mathcal{O}} \in \mathbb{R}^m$ variables. Note that we use the expression $[\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}]$ to vertically concatenate the two column vectors $\zeta_{\mathcal{I}}$ and

$\zeta_{\mathcal{O}}$. The joint probability is formed by superposition of K linear Gaussian functions:

$$\mathcal{P}([\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}]) = \sum_{k=1}^K \pi^k \mathcal{N}([\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}] | \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k) \quad (5)$$

where π^k , $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$ respectively are the prior, mean and covariance matrix of the k -th Gaussian function $\mathcal{N}([\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}] | \boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)$. Given the joint distribution $\mathcal{P}([\zeta_{\mathcal{I}}; \zeta_{\mathcal{O}}])$ and a query point $\zeta_{\mathcal{I}}$, the GMR process consists of taking the posterior mean estimate of the conditional distribution:

$$\zeta_{\mathcal{O}} = \mathbf{f}(\zeta_{\mathcal{I}}) = \sum_{k=1}^K \frac{\pi^k \mathcal{N}(\zeta_{\mathcal{I}} | \boldsymbol{\mu}_{\mathcal{I}}^k, \boldsymbol{\Sigma}_{\mathcal{I}}^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\zeta_{\mathcal{I}} | \boldsymbol{\mu}_{\mathcal{I}}^i, \boldsymbol{\Sigma}_{\mathcal{I}}^i)} \left(\boldsymbol{\mu}_{\mathcal{O}}^k + \boldsymbol{\Sigma}_{\mathcal{O}\mathcal{I}}^k (\boldsymbol{\Sigma}_{\mathcal{I}}^k)^{-1} (\zeta_{\mathcal{I}} - \boldsymbol{\mu}_{\mathcal{I}}^k) \right) \quad (6)$$

where the subscripts $(\cdot)_{\mathcal{I}}$, $(\cdot)_{\mathcal{O}}$, and $(\cdot)_{\mathcal{O}\mathcal{I}}$ respectively refer to the input, output, and cross partitions of $\boldsymbol{\mu}^k$ and $\boldsymbol{\Sigma}^k$.

B. Gaussian Mixture Regression, a DS view

The probabilistic form of GMR as per Eq. (6) can hardly provide insights on general dynamical behavior of \mathbf{f} . In this section, we rewrite Eq. (6) and reformulate it as a PD controller with state-varying gains (also known as gain scheduling). We denote the input and output variables as $\zeta_{\mathcal{I}} = [\mathbf{s}; \dot{\mathbf{s}}]$ and $\zeta_{\mathcal{O}} = \boldsymbol{\tau}$, where $\mathbf{s} \in \mathbb{R}^d$. The notation of Eq. (6) can be simplified through a set of change of variables. Let us consider:

$$h^k(\zeta_{\mathcal{I}}) = h^k(\mathbf{s}, \dot{\mathbf{s}}) = \frac{\pi^k \mathcal{N}(\zeta_{\mathcal{I}} | \boldsymbol{\mu}_{\mathcal{I}}^k, \boldsymbol{\Sigma}_{\mathcal{I}}^k)}{\sum_{i=1}^K \pi^i \mathcal{N}(\zeta_{\mathcal{I}} | \boldsymbol{\mu}_{\mathcal{I}}^i, \boldsymbol{\Sigma}_{\mathcal{I}}^i)} \quad (7a)$$

$$\boldsymbol{\Sigma}_{\mathcal{O}\mathcal{I}}^k (\boldsymbol{\Sigma}_{\mathcal{I}}^k)^{-1} = \begin{bmatrix} -\mathbf{S}^k & \\ & -\mathbf{D}^k \end{bmatrix} \quad (7b)$$

where $0 < h^k(\mathbf{s}, \dot{\mathbf{s}}) \leq 1$ are scalar functions, and \mathbf{S}^k and \mathbf{D}^k are $d \times d$ matrices. Equation (7b) reveals the relation between the covariance structure of the GMM and the spring-dampers of the DS resulting from GMR. Substituting Eq. (7) into Eq. (6) and rearranging yields:

$$\boldsymbol{\tau}_{IC} = \mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) = - \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{S}^k (\mathbf{s} - \boldsymbol{\mu}_{\mathcal{I}}^k) - \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{D}^k (\dot{\mathbf{s}} - \boldsymbol{\mu}_{\dot{\mathcal{I}}}^k) + \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \boldsymbol{\mu}_{\boldsymbol{\tau}}^k \quad (8)$$

If we further enforces that the matrices \mathbf{S}^k and \mathbf{D}^k be positive definite, i.e. $\mathbf{S}^k \succ 0, \mathbf{D}^k \succ 0, \forall k = 1..K$, then Eq. (8) has the following physical interpretation: each matrix \mathbf{S}^k corresponds now to a linear spring that pulls \mathbf{s} towards its fixed point $\boldsymbol{\mu}_{\mathcal{I}}^k$. Similarly, matrices \mathbf{D}^k are linear dampers that damp any velocity component except $\boldsymbol{\mu}_{\dot{\mathcal{I}}}^k$. Finally, each spring \mathbf{S}^k is preloaded by a force $\boldsymbol{\mu}_{\boldsymbol{\tau}}^k$, see the third term in Eq. (8). The nonlinear weights $h^k(\mathbf{s}, \dot{\mathbf{s}})$ determine the contribution of each preloaded spring and damper. In order to compact the notation, we introduce the following:

$$\begin{aligned} \bar{\mathbf{S}}(\mathbf{s}, \dot{\mathbf{s}}) &= \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{S}^k \\ \bar{\mathbf{D}}(\mathbf{s}, \dot{\mathbf{s}}) &= \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{D}^k \\ \bar{\boldsymbol{\mu}}_{\mathbf{s}}(\mathbf{s}, \dot{\mathbf{s}}) &= \bar{\mathbf{S}}(\mathbf{s}, \dot{\mathbf{s}})^{-1} \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{S}^k \boldsymbol{\mu}_{\mathcal{I}}^k \\ \bar{\boldsymbol{\mu}}_{\dot{\mathbf{s}}}(\mathbf{s}, \dot{\mathbf{s}}) &= \bar{\mathbf{D}}(\mathbf{s}, \dot{\mathbf{s}})^{-1} \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \mathbf{D}^k \boldsymbol{\mu}_{\dot{\mathcal{I}}}^k \\ \bar{\mathbf{f}}_{\boldsymbol{\tau}}(\mathbf{s}, \dot{\mathbf{s}}) &= \sum_{k=1}^K h^k(\mathbf{s}, \dot{\mathbf{s}}) \boldsymbol{\mu}_{\boldsymbol{\tau}}^k \end{aligned} \quad (9)$$

where $\bar{\mathbf{S}}(\mathbf{s}, \dot{\mathbf{s}})$ and $\bar{\mathbf{D}}(\mathbf{s}, \dot{\mathbf{s}})$ correspond to effective proportional and derivative gains at each state $(\mathbf{s}, \dot{\mathbf{s}})$, respectively. Furthermore, $\bar{\boldsymbol{\mu}}_{\mathbf{s}}(\mathbf{s}, \dot{\mathbf{s}})$ and $\bar{\boldsymbol{\mu}}_{\dot{\mathbf{s}}}(\mathbf{s}, \dot{\mathbf{s}})$ are fixed points of the effective spring and damper, and $\bar{\mathbf{f}}_{\boldsymbol{\tau}}(\mathbf{s}, \dot{\mathbf{s}})$ stands for the net preloaded force. Note that since all matrices \mathbf{S}^k and \mathbf{D}^k are positive definite, $\bar{\mathbf{S}}$ and $\bar{\mathbf{D}}$ are also positive definite and thus invertible. Substituting Eq. (9) into Eq. (8) yields:

$$\boldsymbol{\tau}_{IC} = \mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) = -\bar{\mathbf{S}}(\mathbf{s}, \dot{\mathbf{s}}) (\mathbf{s} - \bar{\boldsymbol{\mu}}_{\mathbf{s}}(\mathbf{s}, \dot{\mathbf{s}})) - \bar{\mathbf{D}}(\mathbf{s}, \dot{\mathbf{s}}) (\dot{\mathbf{s}} - \bar{\boldsymbol{\mu}}_{\dot{\mathbf{s}}}(\mathbf{s}, \dot{\mathbf{s}})) + \bar{\mathbf{f}}_{\boldsymbol{\tau}}(\mathbf{s}, \dot{\mathbf{s}}) \quad (10)$$

By inspecting Eq. (10), we could now draw some conclusions about the behavior of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$. By assuming positive definiteness of matrices \mathbf{S}^k and \mathbf{D}^k , Eq. (10) can be interpreted as a spring-damper system, where the stiffness of the spring and the viscosity of the damper varies across the state space. In addition, the evolution of the spring equilibrium position and the undamped velocity are varying across the state space. The last term, $\bar{\mathbf{f}}_{\boldsymbol{\tau}}(\mathbf{s}, \dot{\mathbf{s}})$, has a physical interpretation as a state varying spring preload. From another perspective, this equation can also be seen as an impedance control policy with state-varying stiffness and damping matrices subject to a state-varying load.

IV. STABILITY ANALYSIS

The requirement on positive definiteness of matrices \mathbf{S}^k and \mathbf{D}^k is not sufficient to ensure stability of a state-varying impedance controller as per Eq. (10). Figure 2 shows a simple 1D example that illustrates why this is not the case and also why ensuring stability of state-varying impedance control even in 1D is non-trivial. In this section we derive a set of stability conditions on these parameters so as to ensure global asymptotic stability of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ at the target. Without loss of generality, we assume that the target point \mathbf{s}^* is located at the origin and is not moving, i.e. $\mathbf{s}^* = \dot{\mathbf{s}}^* = \mathbf{0}$. Note that by unifying motion generation and impedance control, the notion of tracking a reference trajectory is entirely removed from our approach. Thus, we only need to prove stability for the regulator case.

A. Candidate Lyapunov Function

We use Lyapunov's direct method to ensure global asymptotic stability of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$. Stability analysis using Lyapunov's direct method requires 1) finding a non-negative Lyapunov function $V \geq 0$ (also called energy function), and 2) verifying that it always decreases as the motion evolves and vanishes at the target. Note that the

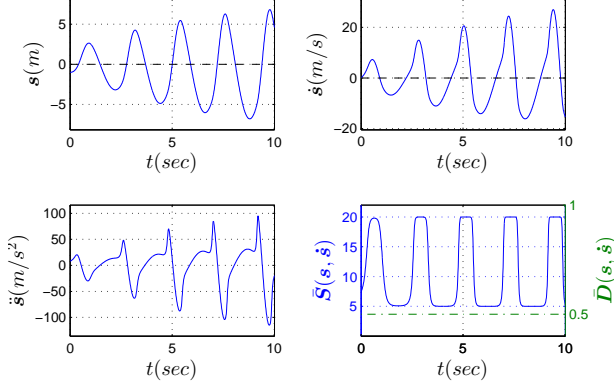


Fig. 2: A one dimensional example showing instability of a variable impedance controller with positive definite stiffness and damping matrices on a point-mass robot with $m = 1kg$. The DS is modeled with the following values: $\pi^1 = 0.75, \pi^2 = 0.25, \mu_T^1 = [1; 4], \mu_T^2 = [-1; -3], \mu_O^1 = \mu_O^2 = 0, S^1 = 20, S^2 = 5, D^T = D^2 = 0.5, \Sigma_T^1 = \Sigma_T^2 = 2I$. The motion starts at $s^0 = -1$ and $\dot{s}^0 = 0$.

conventional way of defining Lyapunov function as sum of potential and kinetic energies of each spring-damper falls short to prove global asymptotic stability of state-varying impedance controller. Hence, in this paper we seek a new way of defining energy by considering the following four requirements:

- 1) As we are interested in ensuring global asymptotic stability of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$, the candidate energy function should have a unique global minimum, located at the target.
- 2) In order to have state-varying stiffness and damping, the candidate energy function should also allow modeling spring-damper systems that are active locally.
- 3) The candidate energy function should have the same parameters as the function $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$. In this way, we can obtain a task-specific energy function without introducing extra parameters.
- 4) The general characteristics of the candidate energy function can be evaluated in closed-form (i.e. without any need to perform numerical check on a mesh of datapoints defined over state-space). This criteria is essential to quickly evaluate stability of the system.

Considering the above four factors, we propose the following candidate Lyapunov function to represent the energy of $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ that is modeled with K Gaussian functions:

$$V(\mathbf{s}, \dot{\mathbf{s}}) = \frac{1}{2} \mathbf{s}^T \mathbf{S}^0 \mathbf{s} + \sum_{k=1}^K \frac{1}{\ell^k} (1 - \beta^k(\mathbf{s})) + \frac{1}{2} \dot{\mathbf{s}}^T \mathbf{M}(\mathbf{s}) \dot{\mathbf{s}} \quad (11)$$

where $(\cdot)^T$ denotes the transpose, \mathbf{S}^0 is the base stiffness and is active globally, $\mathbf{M}(\mathbf{s})$ is the mass matrix, ℓ^k are positive scalars, and $\beta^k(\mathbf{s})$ is given by:

$$\alpha^k(\mathbf{s}) = \begin{cases} \mathbf{s}^T \mathbf{S}^k (\mathbf{s} - 2\boldsymbol{\mu}_s^k) & \text{if } \mathbf{s}^T \mathbf{S}^k (\mathbf{s} - 2\boldsymbol{\mu}_s^k) \geq 0 \\ 0 & \text{if } \mathbf{s}^T \mathbf{S}^k (\mathbf{s} - 2\boldsymbol{\mu}_s^k) < 0 \end{cases} \quad (12)$$

$$\beta^k(\mathbf{s}) = e^{-\frac{\ell^k}{4} (\alpha^k(\mathbf{s}))^2} \quad (13)$$

Figure 3 shows the effect of $\boldsymbol{\mu}_s^k$, \mathbf{S}^k , and ℓ^k on the energy function of a uni-dimensional DS. Figure 4 also illustrates two examples of the energy function for 2D systems. First note that the function $\beta^k(\mathbf{s})$ and by construction $V(\mathbf{s}, \dot{\mathbf{s}})$ are continuous and have continuous first order partial derivatives (i.e. class \mathcal{C}^1 smoothness). Second, $V(\mathbf{s}, \dot{\mathbf{s}})$ is always non-negative provided $\ell^k > 0$. Third, the exponential term in Eq. (13) provides the possibility to have local effect for each spring (we will elaborate later on in this section).

B. Stability Conditions

In this section, we introduce a set of conditions based on our proposed candidate Lyapunov function such that $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}})$ becomes globally asymptotically stable.

Theorem 1 Consider a state-varying impedance control law given by an autonomous DS $\mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) : \mathbb{R}^{d \times d} \mapsto \mathbb{R}^d$:

$$\begin{aligned} \boldsymbol{\tau}_{IC} = \mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) = & -\mathbf{S}^0 \mathbf{s} - \mathbf{D}^0 \dot{\mathbf{s}} - \sum_{k=1}^K \omega^k(\mathbf{s}) \mathbf{S}^k (\mathbf{s} - \boldsymbol{\mu}_s^k) \dots \\ & - \sum_{k=1}^K \omega^k(\mathbf{s}) \mathbf{D}^k (\dot{\mathbf{s}} - \boldsymbol{\mu}_{\dot{\mathbf{s}}}^k) + \sum_{k=1}^K \omega^k(\mathbf{s}) \boldsymbol{\mu}_{\boldsymbol{\tau}}^k \end{aligned} \quad (14)$$

where \mathbf{S}^0 and \mathbf{D}^0 are the base stiffness and damping that are active throughout the motion, and the nonlinear weighting coefficients $\omega^k(\mathbf{s})$ are given by:

$$\omega^k(\mathbf{s}) = \alpha^k(\mathbf{s}) \beta^k(\mathbf{s}) \quad \forall k = 1..K \quad (15)$$

The robot motion driven by the impedance control law $\boldsymbol{\tau} = \mathbf{f}(\mathbf{s}, \dot{\mathbf{s}}) + \mathbf{g}(\mathbf{s})$ is globally asymptotically stable at its unique attractor \mathbf{s}^* if:

$$\mathbf{S}^0 = (\mathbf{S}^0)^T \succ 0 \quad (16a)$$

$$\mathbf{D}^0 \succ 0 \quad (16b)$$

$$\mathbf{S}^k = (\mathbf{S}^k)^T \succeq 0 \quad \forall k = 1..K \quad (16c)$$

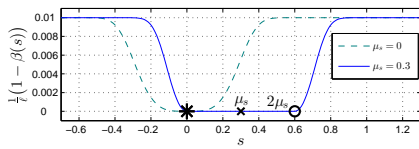
$$\mathbf{D}^k \succeq 0 \quad \forall k = 1..K \quad (16d)$$

$$\ell^k > 0 \quad \forall k = 1..K \quad (16e)$$

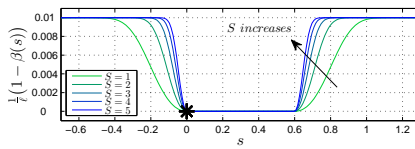
$$\boldsymbol{\mu}_{\dot{\mathbf{s}}}^k = \boldsymbol{\mu}_{\boldsymbol{\tau}}^k = \mathbf{0} \quad \forall k = 1..K \quad (16f)$$

Proof: See Appendix A in the supplementary document.

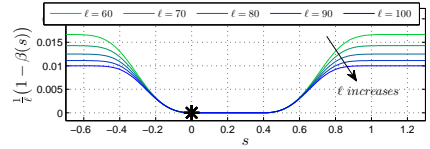
Note that Eq. (14) is similar in structure to the GMR formulation given by Eq. (8) (or its equivalent in Eq. (10)) with the only difference that the nonlinear weights $h^k(\mathbf{s}, \dot{\mathbf{s}})$ are now replaced by $\omega^k(\mathbf{s})$. As opposed to $h^k(\mathbf{s}, \dot{\mathbf{s}})$ in GMR, the weights $\omega^k(\mathbf{s}) \geq 0$ are designed so as to ensure that the accumulative effect of the K spring-damper systems does not generate any spurious attractor, nor cause instability. The constraints given by Eqs. (16a) to (16d) are simply stating that for stability, our state-varying gains should remain positive definite,



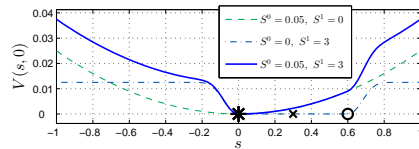
(a) Effect of changing μ_s on an energy component $\frac{1}{\ell^k}(1 - \beta^k(s))$. The following values are used for generation of this figure: $S^1 = 2$ and $\ell^1 = 100$.



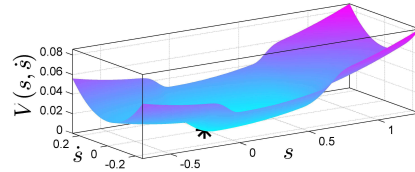
(b) Effect of changing S on the energy component $\frac{1}{\ell^k}(1 - \beta^k(s))$. The following values are used for generation of this figure: $\mu_s^1 = 0.3$ and $\ell^1 = 100$.



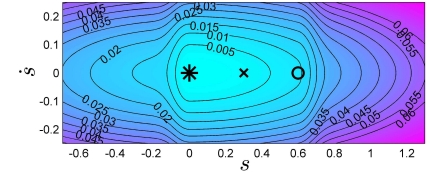
(c) Effect of changing ℓ on the energy component $\frac{1}{\ell^k}(1 - \beta^k(s))$. The following values are used: $\mu_s^1 = 0.3$ and $S^1 = 1$.



(d) Creating the energy function with $K = 1$. The following values are used for generation of this figure: $S^0 = 0.05$, $\mu_s^1 = 0.3$, $S^1 = 3$, $\ell^1 = 80$.

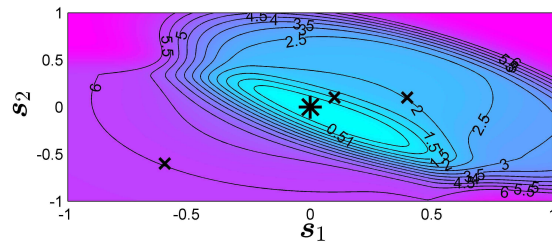


(e) Illustrating the energy function (both position and velocity terms) for the parameters given in (d).

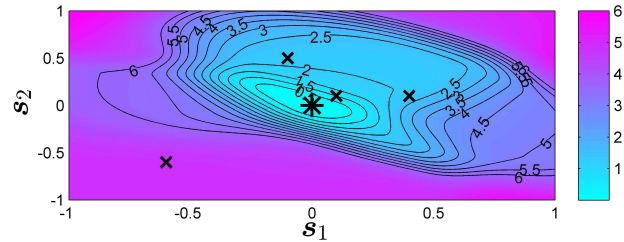


(f) Illustrating the energy levels for the parameters given in (d).

Fig. 3: Evaluating the effect of μ_s^k , S^k , and ℓ^k on the energy function of a uni-dimensional DS. In Fig. 3a, note the zero value of the energy when $s \geq 0$ and $s \leq 2\mu_s$. This is essential to ensure that the accumulation of energy elements $\frac{1}{\ell^k}(1 - \beta^k(s))$ in Eq. (11) does not introduce any local minima, and by construction makes the target point (indicated with a star) the unique global minimum of the energy function regardless of the number of components K and the values of μ_s^k , S^k , and ℓ^k . For clarity of the graph, without loss of generality, the energy function in Fig. 3d is computed only for the positional part, i.e. assumed $\dot{s} = 0$.



(a) An example with $K = 3$, $\mu_s^1 = [0.1; 0.1]$, $\mu_s^2 = [-0.6; 0.6]$, $\mu_s^3 = [0.4; 0.1]$, $S^0 = 0.05I$, $S^1 = [4 \ 0; 0 \ 4]$, $S^2 = [6 \ 2; 2 \ 8]$, $S^3 = [12 \ 8; 8 \ 12]$, $\ell^1 = 2\ell^2 = 5\ell^3 = 1$.



(b) An example with $K = 4$. For $k = 1, 3$, we use the same parameters as in (a). For the fourth element, the following values are used: $\mu_s^4 = [-0.1; 0.5]$, $S^4 = [4 \ -2; -2 \ 7]$, $\ell^4 = 0.5$.

Fig. 4: Illustration of the energy function for two 2D examples. The target point and the spring fixed-points are shown with a star and cross, respectively. Note the energy functions are only depicted for the positional part, i.e. assumed $\dot{s} = 0$ (for $d = 2$, $V(s, \dot{s}) : \mathbb{R}^4 \mapsto \mathbb{R}^+$ which is not possible to illustrate for all situations). As we can see, complex energy functions can be created by varying the number of spring-damper systems K and the value of their parameters.

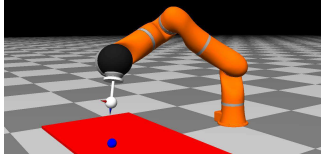
analogous to classical spring-damper systems. Note that the damping matrices need not to be symmetric. The skew-symmetric part of each D^k represents the so-called conservative gyroscopic forces, which transfers energy from one direction to another without any dissipation [27]. Considering Eq. (13), a negative ℓ^k could cause the influence function goes to infinity, yielding unbounded control. The last constraint Eq. (16f) is also essential to avoid energy pumping to the system, which can also cause instability.

Note that Theorem 1 guarantees global asymptotic stability of the robot for free-space motion. As is shown in Appendix B in the supplementary document, Theorem 1 also implies a passive map from external force τ_{ext} to \dot{s} , hence guaranteeing that i-MOGIC remains stable (but not asymptotically stable) in contact with any passive environment.

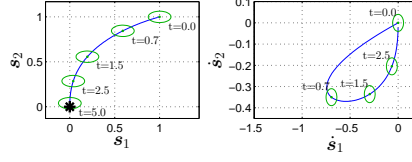
V. SIMULATION RESULTS

In this paper we consider two simulation experiments that are performed on the 7-DoF KUKA LWR arm to highlight different aspects of the proposed approach. In both experiments, control is performed in the operational space formulation [28].

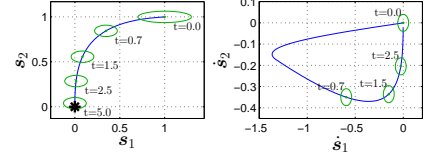
In the first experiment, we show how the user can interactively modify the behavior of the system in two dimension, i.e. $d = 2$. In this example, the first objective is to reach a target point located on a table by following a single curvy movement. The movement is defined in the vertical plane, i.e. $x - z$ with $s_1 = x$, $s_2 = z$, and $y = 0$. The surface of the table is at $s_2 = z = 0$. This movement can be achieved by using a single spring-damper system with $S^0 = [4 \ 0; 0 \ 1]$ and $D^0 = [4 \ 0; 0 \ 2]$ as shown in Fig. 5b. For this system, the stiffness and damping are fixed throughout the motion.



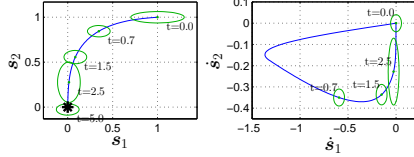
(a) Simulator of the 7-DOF DLR-IV robot. The target point is marked with the blue ball.



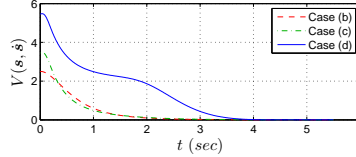
(b) A one-curve reaching movement derived solely by the base DS \mathbf{S}^0 and \mathbf{D}^0 . The initial point is $\mathbf{s} = [1; 1]$ and $\dot{\mathbf{s}} = [0; 0]$



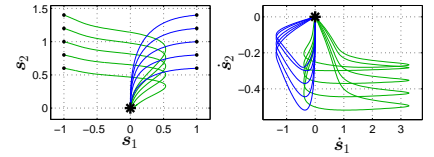
(c) Adding the first spring-damper system to modify both the motion and stiffness at the beginning of the task.



(d) Addition of the second spring-damper system to modify the damping in the region close to the target.



(e) Comparing the energy of the motion between the cases (b), (c), and (d).



(f) Generalization of the model in (d) to different initial conditions.

Fig. 5: In this example the user interactively modifies the behavior of a 2D motion to have different stiffness or trajectory while approaching the target. The effective gains $\tilde{\mathbf{S}}(\mathbf{s}) = \sum_{k=1}^K \omega^k(\mathbf{s})\mathbf{S}^k$ and $\tilde{\mathbf{D}}(\mathbf{s}) = \sum_{k=1}^K \omega^k(\mathbf{s})\mathbf{D}^k$ are shown with superimposed ellipses on the trajectory. The center of each ellipse represents the point at which $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{D}}$ are evaluated.

Now let us consider a situation where it is necessary to achieve a high velocity along the s_1 -axis at the beginning of the motion. To reach this goal we could add, for example, a spring-damper element with the following parameters: $\boldsymbol{\mu}_s^1 = [0.4; 0.0]$, $\ell^1 = 0.1$, $\mathbf{S}^1 = [10 \ 0; 0 \ 0]$, and $\mathbf{D}^1 = \mathbf{0}$. With this element, the velocity increases as long as $s_1 > 2\boldsymbol{\mu}_s = 0.8$. As we are interested in increasing the velocity solely along the s_1 -axis, we set the eigenvalue of \mathbf{S}^1 along s_2 to zero. We also consider $\mathbf{D}^1 = \mathbf{0}$ to achieve a higher acceleration. Figure 5c shows the behavior of the new system. As can be seen the effective gain $\tilde{\mathbf{S}}(\mathbf{s})$ at the beginning of the motion is increased and as a result the velocity reaches a higher peak (about $\dot{s}_1 = -1.4m/s$) compared to Fig. 5b. This example highlights the coupling effect between the motion generation and impedance control.

Now consider another case where it is desired to increase the damping along the s_2 -axis near the target while retaining the velocity profile similar to the previous situation. This behavior could be useful to improve perturbation rejection along the s_2 -axis near the target, where a vertical disturbance could cause hitting the table. For this objective, we add another spring-damper element with $\mathbf{D}^2 = [0 \ 0; 0 \ 25]$. We also consider $\mathbf{S}^2 = [0 \ 0; 0 \ 10]$ to avoid decrease in the velocity due to the added damper. We tune the influence of this element near the target by defining $\boldsymbol{\mu}_s^2 = [0.0; -0.3]$ and $\ell^2 = 0.5$. As can be seen in Fig. 5d, the damping along the vertical axis is increased, while the velocity profile is preserved as closely as possible to Fig. 5c. This example shows the possibility to decouple the two behaviors through proper tuning of the stiffness and damping matrices.

Figure 5e compares the change in the energy of the motion between the above three cases. As can be seen,

addition of each spring-damper system increases the initial energy of the motion. However, the way the energy drops to zero significantly depends on how the spring-damper parameters are defined. For example here the energy of the motion in the second case drops to zero faster than in the first case (even though it starts at a higher value). Figure 5f shows the generalization of the 2D model shown in Fig. 5d to different initial conditions. Since the model is defined in the task space, the motion could follow different behaviors when it starts from different parts of the task space. In this example, the movements that start from the top-right of the target follow a completely different position and velocity profiles, compared to those that start from the top-left. Consequently, they will also have different stiffness and damping profiles along the motion.

In the second experiment we consider a case where two 3D DS are sequenced to create a cyclic pattern. We choose this experiment to highlight ‘the movement primitive’ property of our control policy, i.e. it can be sequenced or superimposed to create new behaviors [29]. The first DS is modeled with the following parameters: $\mathbf{S}^0 = [2 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 1]$ and $\mathbf{D}^0 = \sqrt{4}\mathbf{S}^0$, $\boldsymbol{\mu}_s^1 = [0; -0.1; 0]$, $\ell^1 = 0.5$, $\mathbf{S}^1 = [2 \ 2 \ 0; 2 \ 4 \ 0; 0 \ 0 \ 2]$, and $\mathbf{D}^1 = \sqrt{4}\mathbf{S}^1$. This DS is used to generate the upward motion. The second DS is modeled with: $\mathbf{S}^0 = [1 \ 0 \ 0; 0 \ 3 \ 0; 0 \ 0 \ 1]$ and $\mathbf{D}^0 = \sqrt{4}\mathbf{S}^0$, $\boldsymbol{\mu}_s^1 = [0; -0.1; 0]$, $\ell^1 = 2$, $\mathbf{S}^1 = [4 \ 2 \ 0; 2 \ 6 \ 0; 0 \ 0 \ 2]$, and $\mathbf{D}^1 = \sqrt{4}\mathbf{S}^1$. This DS generates the downward motion.

The first DS switches to the second DS at $\mathbf{s}^s|_{1 \rightarrow 2} = [0.1; 0.3; 0.3]$ with $\dot{\mathbf{s}}^* = [0.1; 0; 0]$. The second DS switches to the first one at $\mathbf{s}^s|_{2 \rightarrow 1} = [0; 0; 0]$ with $\dot{\mathbf{s}}^* = [-0.1; 0; 0]$. For both dynamics $\ddot{\mathbf{s}}^* = \mathbf{0}$. In order to pass the switching points with the desired veloci-

ties, we employ a moving target approach similarly to [30]. Note that to reach a moving target, we need to augment Eq. (3) with the term $C(s, \dot{s})\dot{s}^*$, i.e. defining $\tau = \tau_{IC} + g(s) + C(s, \dot{s})\dot{s}^*$ (see [14] for more details). The motion starts at $s^0 = [-0.2; 0; 0]$ with zero velocity. Figure 6a shows the initial point, switching points, and the position and velocity profiles of the movement. After the first switch, the motion keep moving on the cyclic pattern. Figure 6b shows the time-evolution of position, velocity, acceleration, energy, and effective gains. As can be seen, the position and velocity profiles are continuous. At the switching points, as expected, there is a jump in the acceleration and also the energy function. As we defined the damping matrices in relation to the stiffness matrices, they both follow the same pattern.

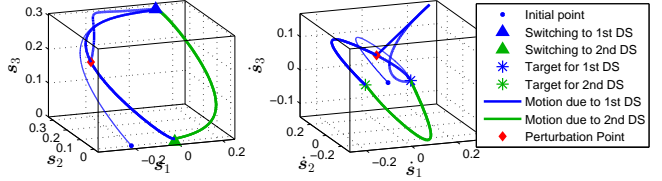
To further investigate the robustness of the system, we perturb the motion at $t = 26\text{sec}$ with the amplitude $\dot{s}^p = [0.3; 0.2; 0.1]$, which is about four times higher than the motion velocity at the time of perturbation. As can be seen, the motion can successfully recover from this perturbation and continue its cyclic behavior. Note that in our approach there is *no reference trajectory*. When perturbed, the motion planner just provides a new path on the fly while still ensuring it reaches the target point.

VI. SUMMARY & CONCLUSION

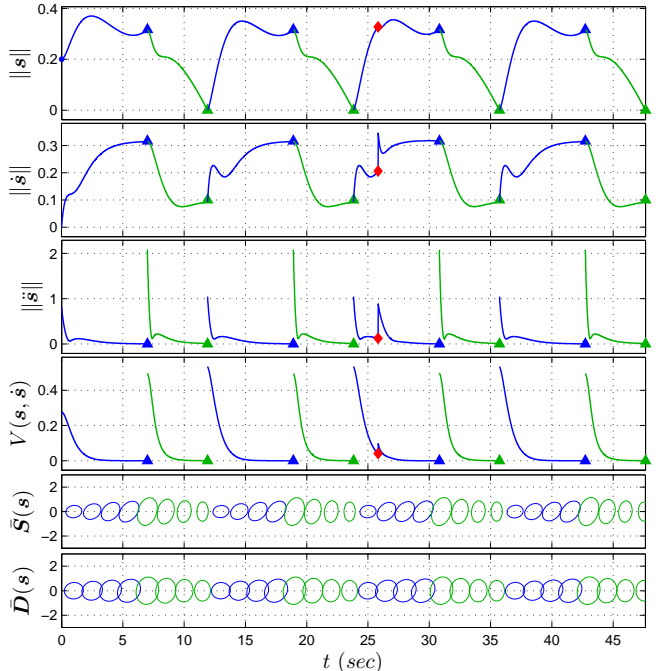
In this paper, we have presented the theoretical foundations of i-MOGIC, an approach to unify motion planning and interaction control into one single control law. I-MOGIC is formulated as a second order autonomous DS and has a similar structure to GMR. It has the following two properties: 1) It can represent both interactive behavior and motion generation, while having a clear physical interpretation, and 2) It is globally (asymptotically) stable. We highlighted these properties through various illustrations and validated it based on two simulation experiments.

The presented approach at its current form is not without its limitations. One possible side-effect of using i-MOGIC is related to the fact that the interaction behavior and the motion generator are determined by the same set of parameters. Although this allows us to ensure stability of the system, it makes it more difficult to modify each of these behaviors independently. Furthermore, the theoretical stability proof of i-MOGIC comes at a cost: i-MOGIC is less flexible in learning complex motions compared to the conventional two-loop control architecture.

The choice of a state-varying (i.e. time-invariant) as opposed to a time-varying impedance controller depends on the task at hand. For many manipulation tasks, the time invariance property is beneficial as it offers robustness in the face of delays during task completion. A time-varying formulation may fail to provide adequate



(a) Position (left) and velocity (right) of the motion.



(b) Illustration of the time-evolution of the position, velocity, acceleration, energy, and effective gains during the motion execution. Note that the ellipses show the projection of $\tilde{S}(s)$ and $\tilde{D}(s)$ in the $s_1 - s_2$ plane.

Fig. 6: Generating cyclic motions with combination of two DS. Please refer to Section V for further information.

impedance parameters at the right time and place if the robot is moving at a slower pace than originally planned, for example, due to inaccuracy in the friction model. However, if the task by itself is time-varying, then the choice of a time-varying impedance controller could be beneficial.

Transfer of skills to the robot including impedance variation information is a field of active research in the robot learning community. Since i-MOGIC is built upon a probabilistic model, with some modifications, existing machine learning techniques (such as the one presented in our previous work [23]) can be exploited to learn an estimate of the control policy from demonstrations.

ACKNOWLEDGMENT

This work was supported by the European Commission through the EU Project AMARSI (FP7-ICT-248311) and the Swiss National Science Foundation through the National Center of Competence in Research Robotics.

REFERENCES

- [1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [2] N. Hogan, "Impedance control: an approach to manipulation," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, 1985.
- [3] L. Villani and J. D. Schutter, *Handbook of Robotics*. Springer Berlin Heidelberg, 2008, ch. Force Control.
- [4] O. Khatib, L. Sentis, and J.-H. Park, "A Unified Framework for Whole-Body Humanoid Robot Control with Multiple Constraints and Contacts," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2008, vol. 44, pp. 303–312.
- [5] B. Siciliano and J.-J. E. Slotine, "A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems," in *ICAR*, 1991.
- [6] N. Hogan and S. P. Buerger, *Robotics and automation handbook*. Boca Raton, FL: CRC, 2005, ch. Impedance and Interaction Control.
- [7] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A unified passivity based control framework for position, torque and impedance control of flexible joint robots," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2007, vol. 28, pp. 5–21.
- [8] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 249–254.
- [9] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, 2013.
- [10] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. Theodorou, and S. Schaal, "Model-free Reinforcement Learning of Impedance Control in Stochastic Environments," *IEEE Trans. on Autonomous Mental Development*, vol. 4, no. 4, pp. 330–341, 2012.
- [11] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable Impedance Control - A Reinforcement Learning Approach," in *Proceedings of Robotics: Science and Systems*, 2010.
- [12] P. Kormushev, S. Calinon, and D. G. Caldwell, "Robot Motor Skill Coordination with EM-based Reinforcement Learning," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010, pp. 3232–3237.
- [13] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer, 2009.
- [14] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, ser. Springer Tracts in Advanced Robotics, 2008, vol. 49.
- [15] K. Kronander and A. Billard, "Online Learning of Varying Stiffness Through Physical Human-Robot Interaction," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.
- [16] D. Mitrovic, S. Klanke, and S. Vijayakumar, "Learning impedance control of antagonistic systems based on stochastic optimization principles," *The International Journal of Robotics Research*, 2010.
- [17] J. Park and H. Cho, "Impedance control with varying stiffness for parallel-link manipulators," in *Proceedings of the 1998 American Control Conference*, vol. 1, 1998, pp. 478–482.
- [18] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, Oct 2011.
- [19] G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "A versatile biomimetic controller for contact tooling and haptic exploration," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2012, pp. 3329–3334.
- [20] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4948–4953.
- [21] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness," in *Proc. of 17th Int. Federation of Automatic Control World Congress*, 2008, pp. 6751–6756.
- [22] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley, 2000.
- [23] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Trans. on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [24] —, "Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.
- [25] S. Vijayakumar and S. Schaal, "Locally Weighted Projection Regression: An O(n) algorithm for incremental real time learning in high dimensional space," in *Proc. of 17th Int. Conf. on Machine Learning (ICML)*, 2000, pp. 1079–1086.
- [26] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. Springer, 2006.
- [27] O. Vahid-Araghi and F. Golnaraghi, *Friction-induced vibration in lead screw drives*. Springer, 2011.

- [28] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. of Robotics and Automation*, vol. 3, pp. 43–53, 1987.
- [29] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [30] J. Kober, K. Mulling, O. Kromer, C. H. Lampert, B. Scholkopf, and J. Peters, "Movement Templates for Learning of Hitting and Batting," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 853–858.
- [31] J. L. Salle and S. Lefschetz, *Stability by Lyapunov direct method*. New York: Academic Press, 1961.
- [32] J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.