Toward Safe Interaction Control with Dynamical Systems

Klas Kronander and Aude Billard

**Abstract**—Autonomous Dynamical Systems (DS) has emerged as an extremely flexible and powerful method for modeling robotic tasks. Task execution of DS models is typically done in an open-loop manner in combination with standard low level controller, e.g. position controller or impedance controller. Such an arrangement has two important drawbacks 1) it is not passive and 2) the DS model can not respond to physical perturbations on the robot body. These are severe limitations tasks with uncertain physical contacts, e.g. object handovers. We propose a novel control architecture that closes the loop around the DS, ensures passivity and allows tuning of the impedance. We evaluate our approach in a comparative study in an uncertain manipulation task with unexpected contact.

I. INTRODUCTION

Dynamical Systems (DS) has emerged as a general and highly flexible means of representing robot motions. It has been demonstrated that many of the proposed DS formulations lend themselves well to learning, both in a supervised setting as well as reinforcement learning. Furthermore, in special cases qualitative properties such convergence to a limit cycle or stability at an attractor point can be ensured regardless of the data provided to the learning algorithm. We believe that the capability of encoding not only a nominal motion plan but also how the robot should respond to perturbations, the DS task representation is very well suited to object handover tasks which are characterized by a high amount of uncertainty and specifically the need to instantly react to non-predictable behavior of a human.

In parallel to the development of DS-based learning systems, the field of robotic manipulation has in recent years seen an revitalized interest in control of mechanical interaction, a topic which largely rests upon foundations of Hogans impedance control formulation [3]. To use a DS task representation with an impedance controller, it is necessary to integrate the DS over time to yield a reference trajectory, see Fig. 1 left. In such a configuration, the reactivity of the DS is used with respect to perturbations that are captured using external sensors, e.g. the pose estimate of a moving target or obstacles. However, the full power of the DS is not used, since the integration of the reference trajectory disallows the DS to react to physical perturbations on the robot. For physical interaction tasks such as hand-overs, such reactivity can be crucial [? ], and it would hence be desirable to instead use a controller that feeds back the actual state of the robot to the DS, see Fig. 1 right.

An important property for controllers interacting with unknown environments is passivity. A controller that ensures a passive relation between external forces and robot velocity will yield stable behavior in free motion and in contact with any passive environment [1]. In this sense, classic impedance control is only passive in the regulation case and the passivity can no longer be ensured if the desired velocity is non-zero. The loss of passivity during tracking is an important drawback of impedance control and a problem that arises in any controller driven by time-indexed reference trajectories. A through analysis of this problem is provided in [4], which advocates to tackle it by encoding tasks using time-independent velocity fields (first order DS) and proposes a controller that ensures tracking converge of the robot to the desired DS. Related work has proposed a similar approach for passivity in the curve tracking problem [2]. These works exploit a time-independent encoding of the task to ensure passivity and energy-efficient, accurate tracking of the DS. The closed-loop dynamics are however rather complicated and the specification of a mechanical impedance becomes non-intuitive.

In this work, we aim to combine the advantages of impedance control and a passive control system without dependency on time. In contrast to [4] and [2] which are based on redistribution of kinetic energy along the desired direction of motion, we propose a control structure which is based on selective dissipation of energy in directions that are irrelevant to the task. As we shall see, this allows to easily tune the mechanical impedance while ensuring passivity.

We evaluate our controller in a robotic reaching task with limited knowledge on 1) the robot dynamics and 2) the environment. We show that the proposed controller has advantages over classical impedance control both in respecting the shape of the desired motion as well as keeping forces low in unexpected contact.
II. Problem Statement

Let \( f(\xi) \) be a Dynamical System describing the a nominal motion plan for a robotic task. The variable \( \xi \) represents a generalized state variable, which could be e.g. robot joint angles or Cartesian position. Any integral curve of \( f \) represents the desired motion of the robot in the absence of perturbations. Consider a dynamics of a RBD with the generalized state variable \( \xi \):

\[
M(\xi)\ddot{\xi} + C(\dot{\xi})\dot{\xi} + g(\xi) = \tau_c + \tau_e
\]

The goal of this work is to design a controller \( \tau_e \) so that Eq. (1) has the following properties:
1) Passivity \((\tau_e, \dot{\xi})\) should be preserved for the controlled system.
2) The controller should dissipate kinetic energy in directions not relevant for the task.
3) It should be possible to vary task-based impedance if the manipulator, e.g. how dynamics defining how external forces \( \tau_e \) affect the velocity \( \dot{\xi} \).

III. Selective Dissipation via Varying Damping

Consider a feedback controller consisting solely of a damping term and a gravity cancellation term:

\[
\tau_c = g(\xi) - D(\xi)\dot{\xi}
\]

where \( D \in \mathbb{R}^{N \times N} \) is some positive semi-definite matrix.

It is easy to show that a the controller in (2) renders the system passive with respect to the input \( \tau_c \), output \( \dot{\xi} \) with the kinetic energy as storage function. This is true for an arbitrarily varying damping, as long as it remains positive semi-definite. We will exploit this fact and construct a varying damping term that dissipates selectively in directions orthogonal to the desired direction of motion given by \( f(\xi) \).

Let \( e_1, \ldots, e_N \) be an orthonormal basis for \( \mathbb{R}^N \) with \( e_1 \) pointing in the desired direction of motion. Let the matrix \( Q(\xi) \in \mathbb{R}^{N \times N} \) be a matrix whose columns are given by \( e_1, \ldots, e_N \). This matrix is a function of the state \( \xi \), since the vectors \( e_1 \) and hence all \( e_1, \ldots, e_N \) depend on \( \xi \) via \( f(\xi) \). We then define the state-varying damping matrix \( D(\xi) \) as follows:

\[
D(\xi) = Q(\xi)\Lambda Q(\xi)^T
\]

where \( \Lambda \) is a diagonal matrix with non-negative values on the diagonal \( \lambda_1, \ldots, \lambda_N \geq 0 \). By adjusting the eigenvalues, different dissipation behaviors can be achieved. For example, setting \( \lambda_1 = 0 \) and \( \lambda_2, \ldots, \lambda_N > 0 \) results in a system that selectively dissipates energy in directions perpendicular to the desired motion. Hence, external work in irrelevant directions is opposed while along the integral curves of \( f(\xi) \) the system is free to move.

IV. Tracking in Conservative DS

While the selective damping in Section III allowed selective energy dissipation, it can not drive the robot forward along the integral curves of \( f \). In order to achieve this, we have to add some driving control to Eq. (2). This can be achieved through rather simple means, provided that the nominal task model \( f \) is the negative gradient of an associate potential function. This is a restricted class of DS will be referred to as conservative vector DS in the rest of this paper. We will use the notation \( f_C \) to denote a conservative DS, and the associated potential function will be denoted \( V_C \).

Consider now a modified controller with negative velocity error feedback:

\[
\tau_e = g(\xi) - D(\xi)(\dot{\xi} - f_C(\xi)) = g(\xi) - D(\xi)\dot{\xi} + \lambda_1 f_C(\xi)
\]

The last equality is due to the fact that \( f_C(\xi) \) is an eigenvector of \( D(\xi) \) as described in Section III. Consider a storage function consisting of the kinetic energy of the manipulator and the \( \lambda_1 \) multiple of the potential function \( V_C \):

\[
W_C(\xi, \dot{\xi}) = \frac{1}{2} \dot{\xi}^T M(\xi)\dot{\xi} + \lambda_1 V_C(\xi)
\]

Taking the time-derivative along the trajectories of (1) with control given by (4) yields:

\[
\dot{W}_C(\xi, \dot{\xi}) = \frac{1}{2} \dot{\xi}^T (M(\xi) - 2C(\xi))\dot{\xi} - \dot{\xi}^T D(\xi)\dot{\xi} + \dot{\xi}^T \tau_e
\]

\[
+ \lambda_1 \dot{\xi}^T f_C(\xi) + \lambda_1 \nabla V_C^T \dot{\xi}
\]

where dependencies of \( M, D \) on \( \xi \) and \( C \) of \( \dot{\xi} \) have been omitted for cleanliness of notation. In Eq. (4) the first term is null due to the skew-symmetry of the matrix \( M - 2C \) and the last to terms cancel because \( f_C(\xi) = -\nabla V_C \). Hence, we have:

\[
\dot{W}_C(\xi, \dot{\xi}) = -\dot{\xi}^T D(\xi)\dot{\xi} + \dot{\xi}^T \tau_e
\]

which proves the passivity of the system.

Unfortunately, only very simple tasks can be encoded with conservative DS. For more general applicability, it is necessary to modify the controller to also handle non-conservative DS.

V. Extension to Non-conservative DS

Let the general DS be decomposed into a conservative part and a non-conservative part:

\[
f(\xi) = f_C(\xi) + f_R(\xi)
\]

where \( f_R \) denotes the non-conservative part. Note that any system can be written on this form, e.g. if no conservative part can be extracted from \( f \) we would simply have \( f_c \equiv 0 \). Now replacing \( f \) for \( f_c \) in Eq. (4) yields the following rate of change of the storage function:

\[
\dot{W}_C = -\dot{\xi}^T D(\xi)\dot{\xi} + \dot{\xi}^T \tau_e + \lambda_1 \dot{z}
\]

where \( z = \dot{\xi}^T f_R(\xi) \) has been introduced. Clearly, passivity is no longer ensured. We can address this problem by introducing an auxiliary state that can temporarily store energy. Essentially, the idea is that energy can be collected in the virtual storage when Eq. (10) is negative, and released when it is positive. This concept, often referred to as energy tanks [6] and has been widely applied e.g. in haptics and telemanipulation.

This fact can be exploited by augmenting the state vector with a virtual state that is capable of storing energy that would
otherwise be lost in dissipation. This stored energy can then be released in order to implement control actions that would be non-passive in the original system without the storage element. This concept, sometime referred to as energy tanks [6] and has been widely applied e.g. in haptics and telemanipulation.

Let $s \in \mathbb{R}$ denote the level of stored energy. We define its dynamics so that it collects energy from $W_C$:

$$
\dot{s} = \alpha(s)\xi^T D\xi - \beta_s(z,s)\lambda_1 z \tag{11}
$$

The scalar functions $\alpha : \mathbb{R} \mapsto \mathbb{R}$ and $\beta_s, \beta_R : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ control the flow of energy between the virtual storage $s$ and the robot, and will be specified later. The level of energy must be taken into account by the control, so that non-passive control action can not be applied when the storage is depleted, i.e. $s = 0$. Therefore, we redefine the robot control command as follows:

$$
\tau_c = g(\xi) - D\dot{\xi} + \lambda_1 f_c(\xi) + \beta_R(z,s)\lambda_1 f_R(\xi) \tag{12}
$$

Now consider the following storage function:

$$
W = W_C + s \tag{13}
$$

Taking the derivative along the trajectories of Eq. (1) with control command given by (19), using skew symmetry of $M-2C$ and $f_C = -\nabla V_C$ yields:

$$
W(\xi, \dot{\xi}) = -(1-\alpha(s))\xi^T D\dot{\xi} + (\beta_R(z,s) - \beta_s(z,s))\lambda_1 z + \xi^T \tau_c \tag{14}
$$

Negativeness of the first two terms is ensured if:

1) $\alpha(s) \leq 1 \quad \forall s$
2) $\beta_R(z,s) - \beta_s(z,s) = 0 \quad \forall z > 0$

For $W$ to be a valid storage function we must furthermore ensure $s > 0$. In addition, $s$ should be bounded, which considering Eq. (15) leads to the following requirements:

1) $\alpha(s) = 0 \quad \forall s \notin [0, \bar{s}[$
2) $\beta_s(z,s) = 0 \quad \forall z > 0$

Let $f(\xi)$ be decomposed into a conservative part and a non-conservative part:

We shall consider an additional state variable $s \in \mathbb{R}$ that represents stored energy. It is a virtual state to which we can assign arbitrary dynamics. We shall consider dynamics coupled with the robot state variables $\xi, \dot{\xi}$ as follows:

$$
\dot{s} = \alpha(s)\xi^T D\dot{\xi} - \beta_s(z,s)\lambda_1 z \tag{15}
$$

Disregarding for the moment the second term in Eq. (15), it is clear that the first term (energy that would otherwise be dissipated) only adds to the virtual storage as long as the latter remains below its upper bound, $s < \bar{s}$. Now turning to the second term of Eq. (15), $\beta_s(z,s)$ should satisfy:

$$
\begin{aligned}
\beta_s(z,s) &= 0 & s \leq 0 \text{ and } z \geq 0 \\
\beta_s(z,s) &= 0 & s \geq \bar{s} \text{ and } z \leq 0 \\
0 &\leq \beta(z,s) & \text{elsewhere}
\end{aligned} \tag{18}
$$

Considering the second term in Eq. (15), it is clear that with $\beta_s$ satisfying Eq. (18), transfer to the virtual storage $(z < 0)$ is only possible as long as $s < \bar{s}$. Conversely, extraction of energy from the storage $(z > 0)$ is only possible as long as $s > 0$. When the storage is depleted, the controller can no longer be allowed to drive the system along $f_R$ if this results in increase the kinetic energy of the system. Therefore, we introduce the scalar function $\beta_R(z,s)$ whose role is to modify the control signal if the storage is depleted.

$$
\tau_c = g(\xi) - D\dot{\xi} + \lambda_1 f_c(\xi) + \beta_R(z,s)\lambda_1 f_R(\xi) \tag{19}
$$

where $\beta_R : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is a scalar function that should satisfy:

$$
\begin{aligned}
\beta_R(z,s) &= \beta_s(z,s) & z \geq 0 \\
\beta_R(z,s) &\geq \beta_s(z,s) & z < 0
\end{aligned} \tag{20}
$$

We are now ready to state the main result of this section: The specifications of the functions $\alpha, \beta_s, \beta_R$ allow some freedom in the design. For examples of these functions, which are used in this work, refer to Appendix

VI. ROBOT EXPERIMENT

In the experiment in Section ??, we showed how a reshaping the dynamics could improve the task model for putting plates into a dish rack. An open-loop integrated reference trajectory from the starting point was used in conjunction with an impedance controller to control the robot during task execution. This controller was hence not passive, and in addition can cause high contact forces in the event of unexpected contact with objects in the environment as shown in Section ??, Here, we revisit the same task in a comparative study between the open-loop approach and the passive controller described Section ??, The experiments in this section use the task model and the same task set up as in Section ??, The experiments are this time conducted on the KUKA LWR 4+ arm in lieu of the Barrett WAM used in Section ??, We do not decouple the dynamics nor use nominal control. Instead we investigate how the control methods compare when only a gravity model of the robot is available.

1) Experimental setup: In Section ??, the attractor of the task DS was placed correctly, so that as the robot reached the attractor, the plate slide into a slot with negligible contact force. In real scenarios, mismatch between environment state and the expected state is unavoidable. To account for this, we conducted three sets of task executions, in each of which the target location of the task DS was offset in different locations behind the real location of the dish rack, see Fig. 2. In each
set of experiments, 5 task executions were carried out using two different controllers described below.

A. The controller from Section V with $\tau = s(0) = 10$, $\lambda_1 = 20$ and $\lambda_2 = \lambda_2 = 200$ is used. The value of $\lambda_1$ was chosen to the minimal value capable of overcoming static joint friction at the point of departure.

B. Openloop trajectory integrated from the initial position of the robot in combination with the impedance controller described in Section ??, the stiffness was set to $K = kI_{3x3}$ with $k = 100$, the minimum value capable of reaching the point of the task in free motion.

All task executions were started somewhere in a small region shown in Fig. 2. Rotational motion of the end-effector was in both cases simply damped by a high amount (4 Ns/rad) which effectively kept the orientation constant during the task execution.

During the task executions, the Cartesian pose and an estimate of the external force was recorded. The estimate of the external force was computed internally in the KUKA controller using the joint torque sensors of the LWR4+ arm. This estimate is available along with position measurements of the joints via the Fast Research Interface (FRI)[5]. A kinematic model was continuously updated in our C++ software implementation of the two controllers, allowing to convert the Cartesian control force to joint torques using Jacobian transpose control (refer to Section ??). The FRI was also used to command torques to the robot at a frequency of 1 kHz.

2) Results: Since very low gains were used, and no inverse dynamics control was applied, it is not expected that either controller would be able to track the nominal motion given by $f$ with good accuracy. This is confirmed in Fig. 4 which plots the nominal and actual trajectories for each setup for each perturbed location of the dish rack. Especially the ’shortcut’ tendency of the impedance controller, which is also consistently obvious in Fig. 5 which plots the estimated contact force during the trials. Setup B consistently impacts the dishrack before setup A, due to the shortcut effect. PICDS has a clear advantage in terms of respecting shape of the desired reaching motion.

In this particular task, the shortcut effect meant that the robot was approaching the rack from the wrong direction, which sometimes lead to interesting final configurations as depicted in Figures ??, 3a and 3b. In each of the three perturbed scenarios, simulation A consistently placed the plate correctly because the pattern of approach was respected, see Figures ?? and 3c.
Fig. 5. The plots show the norm of the estimated external force over time. The raw data are plotted in gray and a temporal average is plotted in black. Each row shows data from the three different perturbed locations of the dish rack. **Left column:** Setup A. **Right column:** Setup B.

As is clear from Fig. 5, setup A also has an advantage over setup B in terms of contact force after impact. At the time of impact, the reference point for setup B has already reached its final point, which is why there is no gradual ramp-up of the contact force as would normally be expected in contact with a timed trajectory. In the second perturbed location (middle plots in Figures 4 and 5), setup B resulted in some of the trials landing in a final configuration on the rack and some of the trials landed in a configuration under the rack. This is visible in the divergence of the trajectories near the end-point in Fig. 4b right, and also the high variance in the final contact force in Fig. 5b right. It should be emphasized that both controllers have been chosen to be as compliant as possible for this task, but the low stiffness is not enough to ensure a low contact force for positioning errors of this magnitude.

VII. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENTS

REFERENCES


