Proceedings of Machine Learning Research vol 242:1–13, 2024

Stable Modular Control via Contraction Theory for Reinforcement Learning

Bing Song BING.SONG@NTU.EDU.SG
HP-NTU Digital Manufacturing Corporate Lab

Jean-Jacques Slotine JJS@MIT.EDU
Nonlinear Systems Laboratory, Massachusetts Institute of Technology

Quang-Cuong Pham CUONG@NTU.EDU.SG
HP-NTU Digital Manufacturing Corporate Lab

Abstract

We propose a novel perspective to integrate control theoretical results with reinforcement learning (RL) for control stability, robustness, and policy transfer: deploying contraction theory for modular architecture design. We leverage the modularity of contraction theory to design the coordinate transformation that can simplify the nonlinear constraints for stability into algebraically solvable ones, yielding linear constraints on the input gradients of control networks. These constraints can be implemented in the control architecture and hence the learning framework remains unchanged, a minimally invasive way to guarantee control stability. We also derive the corresponding theorems to characterize robustness. To mitigate limitations and requirements of dynamic models, we propose a modular control architecture including the coordinate transformation, composite variables, and task space controllers, which is arguably easy to be integrated with hierarchical RL for robot manipulation in unknown environments and improves its performance. We demonstrate our results in two simulated manipulation scenarios. This work suggests the potential of formulating architecture design problems into creating Riemannian spaces paired with contraction metrics.

Keywords: modularity, contraction theory, reinforcement learning, control stability

1. Introduction

Control theoretical results have been deployed in reinforcement learning (RL) to improve control stability, robustness, and generalization for real-world robotic applications, in the way of formulating control theoretical results into optimization problems (Berkenkamp et al., 2017; Moos et al., 2022), incorporating the results in statistical techniques (Mandlekar et al., 2017; Cheng et al., 2019), reducing the gaps between simulation and the real world (Singh et al., 2018), etc. Here we propose a novel perspective: deploying control theoretical results for modular architecture design.

RL targets at the control problems involving variations, uncertainties, unknowns, and non-convexities. It characterizes physical systems with transition functions and formulates optimal control as decision making for Markov Decision Processes (MDPs).

We start with examining the stability of a control policy for an MDP, focusing on the case that the MDP accurately describes a task involving dynamic parameter variations. The parameter distributions in training and testing are identical.

We consider the type of nonlinear system stability known as contraction (Lohmiller and Slotine, 1998). Similar to but distinct from the region of attraction, in a region of contraction, convergence occurs not with respect to a system equilibrium but to a trajectory. Intuitively, all possible states in the region of contraction, including the unseen ones in training, evolve towards some consistent behaviors, as illustrated in Fig. 1 (a). This implies RL agents can optimize the trajectories, while forcing all possible trajectories to converge to each other.

In our parallel paper Song et al., 2023, we show that there exists stability boundaries, the necessary and sufficient conditions for contraction, varying along dynamic parameters. When an RL agent ignores those boundaries, it allows the (local) optimal policy to produce a small fraction of unstable trajectories, possibly small enough to be ignored in empirical studies. Those instabilities can manifest themselves when the dynamic parameters shift, preventing robustness and policy transfer.

This raises the concern for real-world applications, as stability analysis is often omitted from empirical studies and most RL algorithms do not provide theoretical guarantees. On the other hand, concerns persist regarding whether stability guarantees should restrict learning due to conservatism.

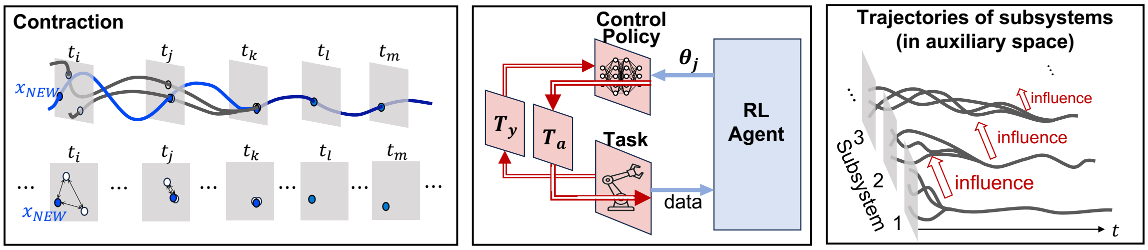
This paper investigates how we may deploy control theoretical results to solve the above concerns, addressing the instabilities in control that prevent policy transfer and robustness.

Deploying control-theoretical results is intrinsically difficult for tasks with large parameter variations. For example, to extend the neural certificates (Dawson et al., 2022) to RL, it requires to understand the interplay between the bounds on the generalization error for the certificates (Boffi et al., 2021), the exploration for value estimation, and task parameter distributions. This interplay has been seldomly investigated in current methods that combine neural certificates with RL, whether through direct approaches like weighted sum of objectives (Qin et al., 2021) and alternative updates (Luo and Ma, 2021), or through mathematically rigorous approaches like unifying the objective functions into a multi-timescale min-max-min optimization problem (Ma et al., 2022). Expertise in both nonlinear control and RL is likely required for the empirical certificate loss design as well as task-specific hyperparameter tuning.

Considering practical examples that involve parameter variations such as contact-rich manipulation and flying in strong winds, learning those skills seems easy and natural to animals. Modularity is believed to be the key in both machine learning (Reed et al., 2022) and nonlinear control, particularly contraction theory (Slotine and Lohmiller, 2001), although the definitions of modularity differ slightly. More than functional specialization, the modularity from contraction theory ensures that combining stable subsystems can automatically preserve the stability (Slotine, 2003).

Seeing the controller synthesis as the combination of designing the control architecture and optimizing neural control policies via RL, we leverage contraction theory to build modular control architectures to guarantee control stability. Firstly, we use coordinate transformation to deconstruct the dynamics by creating an auxiliary space, within which the controlled signals are coupled in a modular structure. We provide one explicit solution of the coordinate transformation that yields hierarchical combinations of subsystems in the auxiliary space, as illustrated in Fig. 1 (b) and (c). Leveraging the modularity of contraction, we simplify the nonlinear constraints into linear ones that can be implemented in the control architecture, yielding a minimally invasive way to guarantee control stability for RL. We also provide theorems to characterize the robustness. Secondly, we use function composition to mitigate the limitations and requirements of dynamics models in deconstructing the dynamics.

Because the learning framework remains unchanged, our approach allows arguably easy integration into the modular architectures in machine learning, in particular, hierarchical RL, and improves its performance by providing consistent low-level dynamic behaviors. Integration with hierarchical RL is also expected to mitigate the concern of conservativeness. Results are demonstrated in two simulated manipulation scenarios. We also briefly include the robustness test of proximal policy



(a) Convergence to consistent behaviors **(b) Control architecture in RL** **(c) Hierarchical type of modularity**

Figure 1: Stable modular control. (a) When a control policy ensures contraction, all possible states, including the unseen ones in training like x_{NEW} at t_i, evolve toward some learned consistent behaviors, the convergence of trajectories. (b) Considering controller synthesis as designing a control architecture (red) and optimizing control policies via RL (blue), the coordinate transformation T_y and T_a creates an auxiliary space, within which the dynamics can be seen as subsystems combined in a modular pattern. (c) The hierarchical type of modularity refers to the combination of subsystems that the lower level affects the higher level in sequence and hence the subsystems converge recursively following their hierarchies, conceptually similar to a robot arm from the base joint to the end effector.

optimization (PPO) (Schulman et al., 2017) in the parallel paper Song et al., 2023. Proofs are in the arXiv preprint at `https://arxiv.org/pdf/2311.03669.pdf`.

Contributions. Our main contribution is the new perspective that guarantees control stability for RL via control architecture design. In particular, we leverage the modularity of contraction theory to design the coordinate transformation that can simplify the nonlinear constraints for stability into algebraically solvable ones, yielding linear constraints on the input gradients of control networks. To our best knowledge, this perspective—constructing modular control architectures via control theoretical results for stability, robustness, and generalization—is new to RL. It is also novel in the control field to design the modular architectures of neural control systems by deconstructing dynamics. This work implies the potential of formulating architecture design problems into creating Riemannian spaces paired with contraction metrics.

Other technical contributions include (i) a modular control architecture that is arguably easy to be integrated in hierarchical RL for robot manipulation in unknown environments and (ii) theorems that characterize the robustness.

2. Related work

Contraction theory in data-driven methodologies. Contraction theory has been used in data-driven methods for robust control (Tsukamoto et al., 2020), adaptive control (Tsukamoto et al., 2021a), motion planning (Tsukamoto and Chung, 2021), and system identification (Singh et al., 2021). Compared with Lyapunov theory, contraction theory analyzes nonlinear systems via their differential dynamics without specifying a system equilibrium, an advantage to analyze the nonlinear systems with uncertain dynamic parameters that shifts the equilibrium (Aylward et al., 2008). A brief comparison of different types of stability can be found in Tsukamoto et al., 2021b.

Safe learning. Brunke et al., 2022 provides an extensive survey on the safe learning. By investigating the stability of a control policy for an MDP, our work focuses on the concern from the fact that shifts in stability boundaries, brought by dynamic parameter variations, can transition a stable policy that is transferable to unseen states into a dangerous one. The results of constraining the input gradients of control networks for stability are in line with Jin and Lavaei, 2020, which focuses on $\mathcal{L}_2$ stability of linear systems with added nonlinearity. In terms of methodology, to guarantee control stability usually involves Step (i) finding a Lyapunov function or a contraction metric and Step (ii) satisfy the nonlinear constraints. Existing methods usually solve Step (i) via searching or learning (Dawson et al., 2022; Sun et al., 2020) and solve Step (ii) by formulating the constraints into optimization problems (Ma et al., 2022; Lale et al., 2022), similarly for approaches deploying control Lyapunov functions (Sontag, 1983) and Control Contraction Metrics (Manchester and Slotine, 2017), the extensions of Lyapunov theory and contraction theory respectively. One interesting direction for RL is leveraging the connection between Lyapunov functions and value functions (Lee and Sutton, 2021; Han et al., 2020; Berkenkamp et al., 2017), as there is a connection between Lyapunov functions and Hamilton-Jacobi-Bellman equations. Here we propose a new perspective that connects contraction metric and control architecture. We design the metric to deconstruct the dynamics. The metric is "implemented" via the coordinate transformation in the architecture to create an auxiliary space, resulting in simplified constraints that can be implemented in control networks.

Modularity Besides motor primitives (Thoroughman and Shadmehr, 2000) and synergies (Santello et al., 2016), modularity can be applied to motion generation in control theory, that is, Riemannian motion policies (Ratliff et al., 2018) pairing different Riemannian metrics to the subsets of the state space defined by different mappings from states to the desired acceleration. Our work shares the spirit in the way that we design the Riemannian metrics to create an auxiliary space for dynamic deconstruction. We also deploy function composition in the architecture, yielding an extra layer of closed-loop control of the latent signals in the latent space. Our work suggests the potential to formulate architecture design problems into creating different Riemannian spaces paired with contraction metrics. In machine learning, modularity (Pfeiffer et al., 2023) is formulated from the perspective of information theory and decision making for functional specialization. The modularity of contraction theory studies how the overall system stability can be automatically guaranteed when combining stable subsystems. Another distinct feature is that contraction theory analyzes the system with differential dynamics. The resulting linear form highly simplifies the analysis.

Related work in robustness and policy transfer. Our parallel paper Song et al., 2023 illustrates that the shifts in stability boundaries, brought by dynamic parameter variations, can transition a stable policy that is transferable to unseen states into a dangerous one, by proving that there exists the stability boundaries varying along dynamic parameters from the necessary and sufficient condition for contraction. It is infeasible to apply the necessary and sufficient condition for stability guarantees, either via contraction theory or Lyapunov-based methodologies. This paper presents how to apply a sufficient condition to deal with the concerns. An extensive discussion on related work in generalization and robustness can be found in Song et al., 2023.

3. Dynamic deconstruction in auxiliary space

We illustrate the general approach and then present the theorem of the explicit solution for the hierarchical type of modularity, followed by the assumptions, limitations, and implementation.

3.1. General approach

Given a nonlinear system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$, we apply the coordinate transformation layers, $\mathbf{z} = T_y \mathbf{y}$ and $\mathbf{u} = T_a \mathbf{a}$, and the neural control policy $\mathbf{a} = \pi(\mathbf{s_1}, \mathbf{s_2})$ where $\mathbf{s}_1 = \mathbf{z}$ and $\mathbf{s}_2 = \int \mathbf{z} dt$. We include $\int \mathbf{z} dt$ in the RL state to force the system equilibrium at the goal for goal tasks, proved in Song et al., 2023. Note that results in this paper hold without $\int \mathbf{z} dt$ and the theorems can be similarly derived by removing the related terms. This yields the following *differential dynamics* in the auxiliary space:

$$\begin{bmatrix} \delta\dot{\mathbf{z}} \\ \delta\dot{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{a} \end{bmatrix} \tag{1}$$

where

$$A = [\dot{T}_y + T_y \frac{\partial \mathbf{f}}{\partial \mathbf{y}}]T_y^{-1}, \; B = T_y \frac{\partial \mathbf{f}}{\partial \mathbf{u}} T_a, \; C = \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{s}_1}(\dot{T}_y + T_y \frac{\partial \mathbf{f}}{\partial \mathbf{y}})T_y^{-1} + \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{s}_2}, \; D = \frac{\partial \boldsymbol{\pi}}{\partial \mathbf{s}_1} T_y \frac{\partial \mathbf{f}}{\partial \mathbf{u}} T_a \tag{2}$$

Without losing generality, we use $\mathbf{z} \in \mathbf{R}^2$ for simplicity.

Each dimension is considered as a subsystem, i.e., subsystems of (z_i, a_i). Rearranging the equations of Eq. 1 by grouping z_i and a_i yields their differential dynamics:

$$\begin{bmatrix} \delta\dot{z}_1 \\ \delta\dot{a}_1 \end{bmatrix} = F_{11} \begin{bmatrix} \delta z_1 \\ \delta a_1 \end{bmatrix} + F_{12} \begin{bmatrix} \delta z_2 \\ \delta a_2 \end{bmatrix}, \quad \begin{bmatrix} \delta\dot{z}_2 \\ \delta\dot{a}_2 \end{bmatrix} = F_{22} \begin{bmatrix} \delta z_2 \\ \delta a_2 \end{bmatrix} + F_{21} \begin{bmatrix} \delta z_1 \\ \delta a_1 \end{bmatrix} \tag{3}$$

where F_{ij} denotes the weight matrices rearranged from Eq. (2). These two subsystems are coupled via the weight matrices F_{12} and F_{21}. The self-feedbacks are F_{11} and F_{12}.

Contraction theory Slotine, 2003 provides and proves the basic types of couplings (combinations) that ensure the overall system preserves the stability, provided each self-feedback is stable. For example, the hierarchical type refers to $F_{12} = 0$ and bounded F_{21}. Intuitively, for the hierarchical type of stability, if each self-feedback loop is stable and the couplings between subsystems are bounded, the subsystems converge recursively following their hierarchies from the lowest to the highest, because the lower subsystem i is always independent from the higher subsystem j for $\forall j > i$, conceptually similar to moving a robot arm from the base joint to the end effector.

To realize a specific type of couplings, one can specify the F_{ij} where $i \neq j$ and solve for T_y and T_a from Eq. (2). The existence of T_y and T_a needs further study. Intuitively, take the hierarchical-type modularity as an example. There are $2n^2$ unknown variables, i.e., the scalar elements of T_y and T_a, to satisfy $2(n^2 - n)$ equations from specifying the coupling matrices, which implies there always exists a solution. Note that the above discussions are with the assumption that $\dot{T}_y$ can be ignored (see Appendix C of the arXiv preprint).

To achieve stable self-feedbacks, one can solve the characteristic equations of those weight matrices F_{ij} where $i = j$ for negative eigenvalues, yielding linear constraints on the input gradients of control networks for stability.

3.2. Stability theorem for the hierarchical-type modularity

Stability theorem (contraction). For a general nonlinear system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$ with diagonalizable $\partial\mathbf{f}/\partial\mathbf{y}$, given a control policy $\boldsymbol{\pi}$, and two transformation layers T_y and T_a that (i) the control commands $\mathbf{u} = T_a \boldsymbol{\pi}(T_y \mathbf{y})$, and (ii) $T_y^T T_y$ and $(T_a^{-1})^T T_a^{-1}$ are uniformly positive definite, if there exists $\alpha > 0$ in a region such that $\forall \mathbf{x}, \forall t > 0$,

$$\frac{\partial \pi^i}{\partial s_{1i}} R_{ii} + \Lambda_{ii} < -\alpha, \quad \frac{\partial \pi^i}{\partial s_{2i}} R_{ii} < -\alpha \tag{4}$$

where R_{ii} and Λ_{ii} are the ith diagonal components of $R = T_y \frac{\partial \mathbf{f}}{\partial \mathbf{u}} T_a$ and $\Lambda = T_y \frac{\partial \mathbf{f}}{\partial \mathbf{y}} T_y^{-1}$ respectively, $\mathbf{s}_1 = T_y \mathbf{y}$ and $\mathbf{s}_2 = \int T_y \mathbf{y} dt$, the neural control system is contracting in the region, provided that

- T_y is the eigenvector matrix from eigenvalue decomposition of diagonalizable $\partial \mathbf{f}/\partial \mathbf{y}$,
- $T_a = (PQ^T)^{-1}$, where Q is the left matrix from the QR decomposition of $PT_y \partial \mathbf{f}/\partial \mathbf{u}$ and P is the permutation matrix with all ones at the skew diagonal,
- each subsystem is controlled by an independent neural network, $\boldsymbol{\pi} = [\pi^1, \cdots, \pi^n]$,
- there exists $N \in \mathbb{R}^+$, $\forall t \geq N$, $\dot{T}_y = 0$, and $\forall t < N$, the dynamics are bounded.

3.3. Robustness

Robustness is the direct outcome of contraction. Considering a task with $\mathbf{f} \in \mathbb{F}$, we can realize contraction for all possible $\mathbf{f}$ by using the inner bounds of the constraints from Eq. (4). This also implies zero-shot policy transfer in terms of a stable controller, while the optimality may be compromised. We provide a theorem to examine the contraction at the presence of model errors in the arXiv preprint. The following is robustness to disturbances. Proofs are also in the arXiv preprint.

Robustness to unknown deterministic perturbation. Considering the perturbed system $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, t) + \mathbf{d}(\mathbf{y}, \mathbf{u}, t)$ with $\mathbf{u} = T_a \boldsymbol{\pi}(T_y \mathbf{y}, \int T_y \mathbf{y} dt)$, where $\mathbf{d}$ represents unknown, bounded, and deterministic disturbances, we rewrite the system with respect to $\bar{\mathbf{y}} = [\mathbf{y}^T, \mathbf{u}^T]^T$, yielding the $\dot{\bar{\mathbf{y}}} = \bar{\mathbf{f}}(\bar{\mathbf{y}}, t) + \bar{\mathbf{d}}$ where $\bar{\mathbf{d}}$ contains all the terms involving $\mathbf{d}(\mathbf{y}, \mathbf{u}, t)$. Let ξ_0 denote a trajectory of the contracting $\dot{\bar{\mathbf{y}}} = \bar{\mathbf{f}}(\bar{\mathbf{y}}, t)$ with the convergence rate β, and ξ_1 a trajectory of the perturbed system $\dot{\bar{\mathbf{y}}} = \bar{\mathbf{f}}(\bar{\mathbf{y}}, t) + \bar{\mathbf{d}}$. The distance between the trajectories of the contracting system and its perturbed dynamics is converging to a bounded error ball:

$$\|\xi_0(t) - \xi_1(t)\| \leq C_1 e^{-\beta t} + C_2 \sup_{\mathbf{y}, \mathbf{u}, t} \|\Theta_M \bar{\mathbf{d}}\| \frac{1 - e^{-\beta t}}{\beta} \tag{5}$$

where Θ_M is the diagonal block matrix with T_y and T_a^{-1} on the diagonal, and C_i's are constants determined by the initial conditions $\xi_i(0)$'s and the bounds of $M = \Theta_M^T \Theta_M$, T_a, T_y, $\partial \boldsymbol{\pi}/\partial \mathbf{s}_1$.

3.4. Assumptions, limitation, and implementation

We assume, for the dynamics $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$, that (i) the locally linearized model, $\partial \mathbf{f}/\partial \mathbf{x}$ and $\partial \mathbf{f}/\partial \mathbf{u}$, is known and (ii) the y and u have the same dimensions.

One limitation is that T_y and T_a needs to be updated step-wise by solving eigenvalue decomposition problems, which can be computational expensive when $\mathbf{y}$ has large dimensions. The other limitation is the conservativeness, coming from two sources. The contraction metric is designed to create modularity instead of minimizing possible conservativeness and the inner bounds of the constraints for all possible dynamic parameters are applied. We expect that integration with hierarchical RL can mitigate the conservativeness, while further studies are needed.

Above assumptions and limitation can be possibly mitigated via creating the latent space by function composition and existing control techniques, illustrated in the next section, the control architecture for hierarchical RL.

Implementation of the stability theorem includes (i) adding two layers T_y and T_a in the control framework, the values of which are updated time-stepwise, and (ii) limiting the network Jacobians $\partial \pi^i / \partial \mathbf{s}_i$ $(i = 1, 2, \cdots, m)$ with the estimated inner bounds of the linear inequality constraints in Eq. (4). We have the following remarks. If $\partial \mathbf{f} / \partial \mathbf{x}$ has negative eigenvalues, that is, $\Lambda_{ii} < 0$, the constraints become switching signs according to R_{ii}. For robotics, negative eigenvalues can be realized by existing controllers and the neural control policy adds an extra layer of closed-loop control (see next section). Further studies are needed to understand the physical implications of R_{ii} and evaluate if and when frequent switching may happen, which likely causes problems. In our experiments, we do not observe any concerns. The stability theorem is derived with continuous dynamic models which applies to low-level control in practice that has fast enough control frequencies. If considering discretization like for high-level planning, the input gradients are likely bounded from two sides. Although the stability theorem does not make any assumptions about the NN architecture, to further simplify the constraints, one can use the activation functions with nonnegative first derivatives. Further simplification can be found in Section V. B. of the arXiv preprint.

4. Modular Control architecture for Hierarchical RL

We propose the modular control architecture in Fig. 2. Besides the coordinate transformation, we introduce a controller $G(\mathbf{x}, \mathbf{x}_d)$ embedded in robots $\dot{\mathbf{x}} = \mathbf{f}_{task}(\mathbf{x}, G(\mathbf{x}, \mathbf{x}_d), \mathbf{u})$ and a composite variable $\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{x}_d)$ into the architecture. The high-level planning running with a lower frequency reads the robot state and plans the desired state over some horizon. The low-level control applies to the composite variable $\mathbf{y}$ within the latent space, the dynamics of which results from the embedded G and g. By doing so, we can leverage control theoretical results to design the embedded controller and the composite variable in the way to mitigate limitations and requirements of dynamic models.

We propose an example using task space controllers (Nakanishi et al., 2008) and composite variables (Slotine, 2003) for robot manipulation in unknown environments. As for the resulting $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$, its dynamic parameters are made of the weights of the task space controller and the composite variables with a diagonal $\partial \mathbf{f} / \partial \mathbf{x}$. Details are in the arXiv preprint.

The low-level neural policy adds an extra layer of closed-loop control in the way to regulate the composite variable, i.e., the latent signal $\mathbf{y}$. The latent state $\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{x}_d)$ converges towards $\mathbf{g}(\mathbf{x}_d, \mathbf{x}_d)$ and the robot state $\mathbf{x}$ converges to $\mathbf{x}_d$. The robot manipulation experiment shows that this extra layer improves the performance of hierarchical RL.

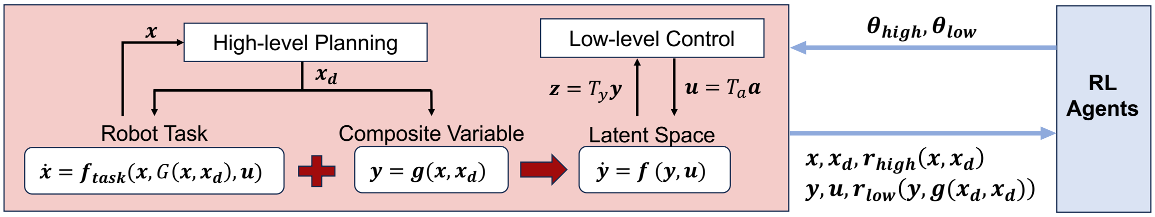


Figure 2: Modular Control architecture for Hierarchical RL.

Considering the auxiliary space, there exists a differential relationship $\delta \mathbf{z} = T_y \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \delta \mathbf{x} + T_y \frac{\partial \mathbf{g}}{\partial \mathbf{x}_d} \delta \mathbf{x}_d$, suggesting that for the state space, there exists a contraction metric made of $\mathbf{g}$, T_y, and T_a. This raises an interesting question if there exists a general connection between contraction metrics and the architecture design techniques, e.g., function composition and coordinate transformation. Can we formulate the architecture design problem into creating different spaces paired with contraction metrics? Another interesting question is that how we may apply this framework to high-dimensional systems using learned latent signals in machine learning. Even more general, this work suggests the potential to design learning system architectures via contraction theory.

5. Example I: Necessity for robustness and generalization

This section briefly presents the robustness and generalization test in the parallel paper Song et al., 2023. The environment approximately simulates a 2D stiff robot (peg-like) touching elastic surfaces (the sketch in Fig. 3), the dynamics of which are described by the following equations

$$\tau_x \dot{x} + x = u_x, \;\; \tau_z \dot{z} + z = u_z, \;\; f = K_s \min\left(z - g(x),\, 0\right), \;\; g(x) = K_1 \sin x + K_2 \cos x \qquad (6)$$

where x and z denotes the position, f represents the force following the Hooke's law, and $g(x)$ is the surface profile along the z-axis. Based on those continuous equations, we built the simulator using the Euler method for discretization with a sampling period T. The task is to move this 2D "peg" to touch a surface at randomly sampled desired position x_d with the desired force f_d by PPO.

Both PPO and the constrained PPO (C-PPO) can learn the tasks with high accuracy, with stable learning curves at similar return levels. Examining 8000 trajectories, we observed 8 oscillating ones, a small fraction $\sim 0.01\%$ of instabilities from PPO. In the robustness test with a larger range of dynamic parameters, the performance of PPO is obviously deteriorated, while C-PPO the preserves the tracking accuracy, better than PPO by two orders of magnitude, illustrated in Fig. 3.

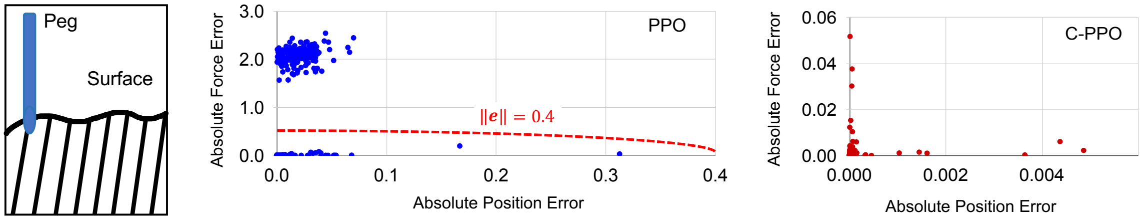


Figure 3: Necessity for robustness and generalization. Policies are tested for 8000 trajectories, with new task parameters with 50% differences in τ_i $(i = x, z)$, 100% in K_{sur} and 50% in K_i $(i = 1, 2)$. In training, the PPO policy produces $\sim 0.01\%$ of trajectories that are oscillating with $\|\mathbf{e}\| > 0.4$ and the largest error is around 1. The middle figure plots the results in testing. The amount of trajectories with $\|\mathbf{e}\| > 0.4$ is increased from 8 to 163 and most of them have the tracking error around 2. C-PPO policy preserves the tracking accuracy, better than PPO by two orders of magnitude.

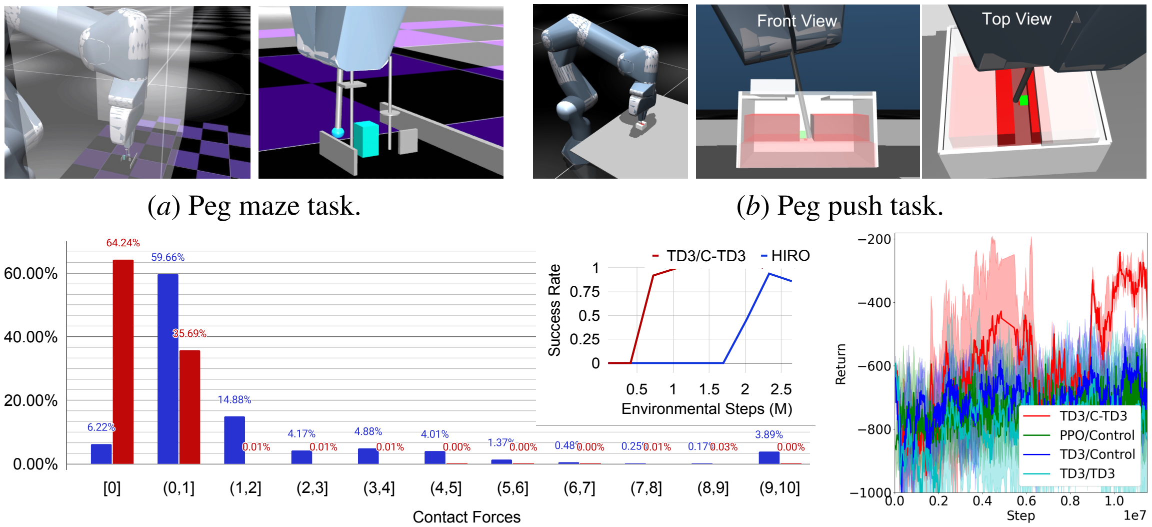
6. Example II: Manipulation learning via hierarchical RL

To implement the modular control architecture for hierarchical RL, we modified the hierarchical learning framework from HIRO (Nachum et al., 2018) in Tensorflow Model Garden. HIRO has outperformed other methods on ant push and ant maze with added relabeling processes to alleviate the non-stationary issue, particularly in data efficiency (Nachum et al., 2018). We extracted the 2-level learning framework using two TD3 agents, removed the relabeling process in HIRO, used the task space instead of the joint space for planning and tracking, added the composite errors, task space controllers, and the coordinate transformation for the low-level control in the architecture.

To evaluate our method, we created peg-maze and peg-push tasks following the concepts of ant maze and ant push. A Franka Panda robot is used to move the peg. The experiments are in Mujoco simulation. Denoting our approach as TD3/C-TD3, we compare our TD3/C-TD3 and HIRO on the easier peg-maze task. With the more difficult peg-push task, an ablation study is performed to evaluate the effect from neural controllers and from stability constraints. We compare our TD3/C-TD3 with (i) one-level learning with non-neural control: TD3/Control and PPO/Control, and (ii) hierarchical RL without stability constraints: TD3/TD3.

Peg maze: TD3/C-TD3 vs. HIRO. The task is to reach a goal in the green zone through the unstructured environment with fixed obstacles (Fig. 4(a)). Robot's motion is limited by a hood. Particularly, the vertical limit forces the peg going through the unstructured zone. Both HIRO and TD3/C-TD3 can find the path with high success rates and with high data efficiency given the initial exploration in the direction of the maze. To evaluate the safety in exploration, we tested policies across the learning curves and compare the distribution of contact forces, plotted in Fig. 4(c). In TD3/C-TD3, 99.93% steps are with less-than-1 N contacts, including 64.24% steps without contacts. In HIRO, 65.88% steps are with less-than-1 N contacts, including 6.22% zero contacts. This shows that TD3/C-TD3 can explore through the obstacles with soft touches, i.e., small contact forces. Besides consistent low-level behaviors, TD3/C-TD3 also benefits from an easier high-level problem, finding a path, compared to planning joint positions in HIRO. Hyperparameters and task paramters are listed in Appendix G of the arXiv preprint.

Peg push: ablation study. The peg-push task is to reach a hidden goal in a box (Fig. 4(b)) that requires interactions with moving objects. We firstly tuned the task space controller, the high-level action range, the low-level horizon, and the fixed trajectory length to make sure that the robot can stably explore the entire environment. The average learning curve from 5 seeds are presented in Fig. 4(d). Only TD3/C-TD3 is able to learn the task (expected return in $[-400, -200]$), and 2 trials have converged after 3M experiment steps. This causes the large variance around 3M to 6M in the averaged learning curve. Other methods fail to learn the task (returns ≈ -700). Close examination of the trajectories shows that TD3/control and PPO/control can learn stable trajectories but haven't figured out the entire path to the goal within 12M steps. Without stability constraints, TD3/TD3 learns unstable trajectories passing by the goal. There still exists one question: whether the soft contact model highly simplifies the task as we observed penetration (the penetration issue discussed in (Parmar et al., 2021)). Further study will focus on evaluation of the neural controller for complex contact stiffness that varies across different surface materials and changes along with relative motions. Appendix H of the arXiv preprint lists the geometry and contact settings of the Mujoco environment, controller gains, RL hyperparameters, etc.



(*a*) Peg maze task.

(*b*) Peg push task.

(*c*) Maze: Contact force distributions and the success rates.

(*d*) Push: learning curves.

Figure 4: Effectiveness in manipulation learning. (a) Peg maze is a goal reaching task with fixed obstacles and limited workspace. (b) Peg push is a goal reaching task that requires opening a sliding lid box, inserting into a deep slot, and pushing away the green obstacle to reach the small red spot underneath. (c) Both TD3/C-TD3 and HIRO can quickly learn the task, $\sim$1M and 2M steps respectively. The contact force distribution are estimated from policies at 10k, 200k, 400k, 600k, 800k, and 1M steps for TD3/C-TD3, and at 10k, 500k, 1M, 1.5M, 2M, and 2.3M steps for HIRO, the success rates of which are plotted in the top right figure. The distribution shows improvements in safe exploration that in TD3/C-TD3, 99.93% are with less than 1 N, and 65.88% with HIRO. (d) Learning curves are plotted. Only TD3/C-TD3 successfully learn the task within 12 M steps, with 2 trials learned the task around 3 M steps causing the large variation. TD3/control and PPO/control can learn stable trajectories but haven't figured out the entire path to the goal. Without stability constraints, TD3/TD3 learns unstable trajectories passing by the goal. Trajectory examples are at https://www.youtube.com/watch?v=nFLHwVfPIJw and https://www.youtube.com/watch?v=RpGzpZPifUw.

7. Concluding Remarks

We propose modular control architectures to improve the control stability, robustness, and policy transfer of RL. This paper illustrates how to leverage the modularity of contraction theory to design the coordinate transformation that makes the signals in an auxiliary space combined in a modular pattern, deconstructing the nonlinear constraints into linear ones. We build a modular control architecture via coordinate transformation, composite variables, and task space controllers for robot manipulation in unknown environments, which is arguably easy to be integrated with hierarchical RL and improves its performance. This work suggests the potential to formulate architecture design into creating Riemannian spaces paired with contraction metrics.

Acknowledgments

This study was supported in part under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner, HP Inc., through the HP-NTU Digital Manufacturing Corporate Lab..

References

Erin M Aylward, Pablo A Parrilo, and Jean-Jacques E Slotine. Stability and robustness analysis of nonlinear systems via contraction metrics and sos programming. *Automatica*, 44(8):2163–2170, 2008.

Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551*, 2017.

Nicholas Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques Slotine, and Vikas Sindhwani. Learning stability certificates from data. In *Conference on Robot Learning*, pages 1341–1350. PMLR, 2021.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Richard Cheng, Abhinav Verma, Gabor Orosz, Swarat Chaudhuri, Yisong Yue, and Joel Burdick. Control regularization for reduced variance reinforcement learning. In *International Conference on Machine Learning*, pages 1141–1150. PMLR, 2019.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods. *arXiv preprint arXiv:2202.11762*, 2022.

Minghao Han, Lixian Zhang, Jun Wang, and Wei Pan. Actor-critic reinforcement learning for control with stability guarantee. *IEEE Robotics and Automation Letters*, 5(4):6217–6224, 2020.

Ming Jin and Javad Lavaei. Stability-certified reinforcement learning: A control-theoretic perspective. *IEEE Access*, 8:229086–229100, 2020.

Sahin Lale, Yuanyuan Shi, Guannan Qu, Kamyar Azizzadenesheli, Adam Wierman, and Anima Anandkumar. Kcrl: Krasovskii-constrained reinforcement learning with guaranteed stability in nonlinear dynamical systems. *arXiv preprint arXiv:2206.01704*, 2022.

Jaeyoung Lee and Richard S Sutton. Policy iterations for reinforcement learning problems in continuous time and space—fundamental theory and methods. *Automatica*, 126:109421, 2021.

Winfried Lohmiller and Jean-Jacques E Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

Yuping Luo and Tengyu Ma. Learning barrier certificates: Towards safe reinforcement learning with zero training-time violations. *Advances in Neural Information Processing Systems*, 34:25621–25632, 2021.

Haitong Ma, Changliu Liu, Shengbo Eben Li, Sifa Zheng, and Jianyu Chen. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In *Learning for Dynamics and Control Conference*, pages 97–109. PMLR, 2022.

Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939. IEEE, 2017.

Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. Robust reinforcement learning: A review of foundations and recent advances. *Machine Learning and Knowledge Extraction*, 4(1):276–315, 2022.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6): 737–757, 2008.

Mihir Parmar, Mathew Halm, and Michael Posa. Fundamental challenges in deep learning for stiff contact dynamics. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5181–5188. IEEE, 2021.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning. *arXiv preprint arXiv:2302.11529*, 2023.

Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates. *arXiv preprint arXiv:2101.05436*, 2021.

Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Marco Santello, Matteo Bianchi, Marco Gabiccini, Emiliano Ricciardi, Gionata Salvietti, Domenico Prattichizzo, Marc Ernst, Alessandro Moscatelli, Henrik Jörntell, Astrid ML Kappers, et al. Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Physics of life reviews*, 17:1–23, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sumeet Singh, Vikas Sindhwani, Jean-Jacques E Slotine, and Marco Pavone. Learning stabilizable dynamical systems via control contraction metrics. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 179–195. Springer, 2018.

Sumeet Singh, Spencer M Richards, Vikas Sindhwani, Jean-Jacques E Slotine, and Marco Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. *The International Journal of Robotics Research*, 40(10-11):1123–1150, 2021.

J-JE Slotine and Winfried Lohmiller. Modularity, evolution, and the binding problem: a view from stability theory. *Neural networks*, 14(2):137–145, 2001.

Jean-Jacques E Slotine. Modular stability tools for distributed computation and control. *International Journal of Adaptive Control and Signal Processing*, 17(6):397–416, 2003.

Bing Song, Jean-Jacques Slotine, and Quang-Cuong Pham. Example when local optimal policies contain unstable control. *arXiv preprint arXiv:2209.07324*, 2023.

Eduardo D Sontag. A lyapunov-like characterization of asymptotic controllability. *SIAM journal on control and optimization*, 21(3):462–471, 1983.

Dawei Sun, Susmit Jha, and Chuchu Fan. Learning certified control using contraction metric. *arXiv preprint arXiv:2011.12569*, 2020.

Kurt A Thoroughman and Reza Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407(6805):742–747, 2000.

Hiroyasu Tsukamoto and Soon-Jo Chung. Learning-based robust motion planning with guaranteed stability: A contraction theory approach. *IEEE Robotics and Automation Letters*, 6(4):6164–6171, 2021.

Hiroyasu Tsukamoto, Soon-Jo Chung, and Jean-Jacques E Slotine. Neural stochastic contraction metrics for learning-based control and estimation. *IEEE Control Systems Letters*, 5(5):1825–1830, 2020.

Hiroyasu Tsukamoto, Soon-Jo Chung, and Jean-Jacques Slotine. Learning-based adaptive control via contraction theory. In *IEEE CDC*, 2021a.

Hiroyasu Tsukamoto, Soon-Jo Chung, and Jean-Jaques E Slotine. Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview. *Annual Reviews in Control*, 2021b.