Structured Policy Representation: Imposing Stability in arbitrarily conditioned dynamic systems

Julen Urain * **Davide Tateo** * **Tianyu Ren** * **Jan Peters** *

Intelligent Autonomous Systems, Technische Universität Darmstadt

Abstract

We present a new family of deep neural network-based dynamic systems. The presented dynamics are globally stable and can be conditioned with an arbitrary context state. We show how these dynamics can be used as structured robot policies. Global stability is one of the most important and straightforward inductive biases as it allows us to impose reasonable behaviors outside the region of the demonstrations.

1 Introduction

Stability is a major concern when the learned policy is running on a real system, where it is crucial to ensure that the generated trajectories converge to a well-defined attractor e.g., a target position or a limit cycle. In most of the robotics manipulation tasks, the robot will perform a trajectory that could be thought of as a trajectory sampled from a stable dynamic system. This is the case of peg-in-a-hole, pouring, or opening a door.

We considered the problem of representing a family of stable policies. These policies could be applied rather in an Imitation Learning (Schaal [1997]) or a Reinforcement Learning (Peters and Schaal [2008]) scenario. Imposing stability in a policy could be beneficial not only for safety issues but also in a learning scenario. By designing inherently stable policies we impose a prior that helps in the generalization, particularly in areas that are far from the previously experienced demonstrations or not explored by the agent.

In our work, we present a new family of globally stable stochastic policies that could represent go-to or cyclic motions. Our proposed policy structure takes advantage of the Conditioned Invertible Neural Networks (INN), to learn rich and highly nonlinear stochastic stable dynamics that could be applied as robot policies. Our work extends Urain et al. [2020], by considering a richer set of INN, deeper stability analysis, and adding arbitrary conditioning.

2 From Policies to Dynamic Systems

A policy π, is a function that maps a state $\boldsymbol{s}$ to a probability distribution over the action $\boldsymbol{a} \sim \pi(\boldsymbol{a}|\boldsymbol{s})$.

In Robotic Manipulation, the action is usually the robot's joint torques($\boldsymbol{\tau}$), and the state is a combination of joint positions, velocities($\boldsymbol{q}, \dot{\boldsymbol{q}}$), and some sensor inputs($\boldsymbol{c}$). As commercial robots often provide access to the dynamic model, we can write mathematically how the joint torques affects the robot's joint state

$$\boldsymbol{\tau} + \boldsymbol{\tau}_{ext} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}), \tag{1}$$

*Correspondence to {urain, tateo, ren, peters}@ias.informatik.tu-darmstadt.de

NeurIPS 2020 3rd Robot Learning Workshop: Grounding Machine Learning Development in the Real World.

where $M(\boldsymbol{q})$ is the joint-space inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the matrix related with the coriolis and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q})$ is the joint-space gravity force vector and $\boldsymbol{\tau}_{ext}$ is the joint-space external torque forces. Given the robot dynamics are known, we can select a policy that cancels out the robot dynamics and depends exclusively in a certain desired acceleration $\ddot{\boldsymbol{q}}_{des}$, (Hogan [1984])

$$\boldsymbol{\tau} = C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + M(\boldsymbol{q})\ddot{\boldsymbol{q}}_{des}. \tag{2}$$

Combining Eq. 1 and Eq. 2,

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_{des} + M^{-1}(\boldsymbol{q})(\boldsymbol{\tau}_{ext}); \tag{3}$$

robot's joint dynamics depends only on our desired acceleration and the external torques. This policy structure is useful for robotic manipulation as we can abstract our policy from torque-space to acceleration-space

$$\ddot{\boldsymbol{q}}_{des} \sim \pi(\ddot{\boldsymbol{q}}_{des}|\dot{\boldsymbol{q}}, \boldsymbol{q}, \boldsymbol{c}). \tag{4}$$

In this context, (4) represents a stochastic dynamic system in the robot's joint-space, conditioned on some context information $\boldsymbol{c}$. Therefore, we can frame our policy as a stochastic dynamic system and impose properties such as stability on it.

3 Stable dynamics with Invertible Neural Networks

In recent years, Normalizing Flows (NF) (Rezende and Mohamed [2015]) have gained popularity as powerful networks for density estimation, variational inference, and generative modeling. NF are explicit likelihood models that construct flexible probability distributions of high-dimensional data composed of a latent distribution from which is easy to sample, $\boldsymbol{z} \sim p(\boldsymbol{z})$, usually a normal distribution; and a INN, $\boldsymbol{y} = f(\boldsymbol{z})$, that maps the latent space $\boldsymbol{Z}$ to observation space $\boldsymbol{Y}$. By the use of change of variable rule, the density on the observation space can be computed in terms of the density in latent space and the invertible transformation's jacobian

$$p(\boldsymbol{y}) = p(\boldsymbol{z}) \left| \det \frac{\partial f}{\partial \boldsymbol{z}} \right|^{-1} \tag{5}$$

NF can be extended to learn dynamic data distributions, by switching the latent static distribution with a dynamic one. Given a latent stochastic dynamic system and an invertible transformation

$$\boldsymbol{z}_{k-1} = f^{-1}(\boldsymbol{y}_{k-1}) \qquad \boldsymbol{z}_k \sim p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) \qquad \boldsymbol{y}_k = f(\boldsymbol{z}_k), \tag{6}$$

the stochastic dynamics in $\boldsymbol{Y}$ can be computed in terms of the stochastic dynamics in $\boldsymbol{Z}$

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{k-1}) = p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) \left| \det \frac{\partial f}{\partial \boldsymbol{z}_k} \right|^{-1}. \tag{7}$$

Stability We have considered the family of stochastic dynamic systems represented with the model proposed in (6). Given the latent dynamics, $p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1})$, are globally stochastically stable, we prove in the Appendix A, that $p(\boldsymbol{y}_k|\boldsymbol{y}_{k-1})$ will also be globally stochastically stable.

Conditioning As presented in (4), a desired robotic policy should be able to consider an arbitrary context state, $\boldsymbol{c}$. Anyway, the dynamics we have presented in (7) can only consider robot's joints. We have switched INN with Conditioned INN to adapt the dynamics to the context state $\boldsymbol{c}$

$$p(\boldsymbol{y}_k|\boldsymbol{y}_{k-1}, \boldsymbol{c}_k) = p(\boldsymbol{z}_k|\boldsymbol{z}_{k-1}) \left| \det \frac{\partial f(\boldsymbol{z}_k, \boldsymbol{c}_k)}{\partial \boldsymbol{z}_k} \right|^{-1}. \tag{8}$$

Given that stability is imposed in the structure of the neural network; our model will remain stable for any arbitrary context. Moreover, while most of the previous works (Khansari-Zadeh and Billard [2011], Calinon [2016], Sindhwani et al. [2018]), only considered affine context-based transformations(different goal target, different velocities ...), our model can learn any nonlinear context-based transformation while remaining stable.

4 Experiments

We have considered three experiments to validate our model. In the first experiment, we considered a behavioral cloning problem on which a highly nonlinear limit cycle is learned from data. In the second experiment, we wanted to evaluate the performance of our policies when increasing the dimensionality of an arbitrary context. Finally, in the last experiment, we evaluate the performance of our model in a real robot scenario, with switching contexts and external perturbances.

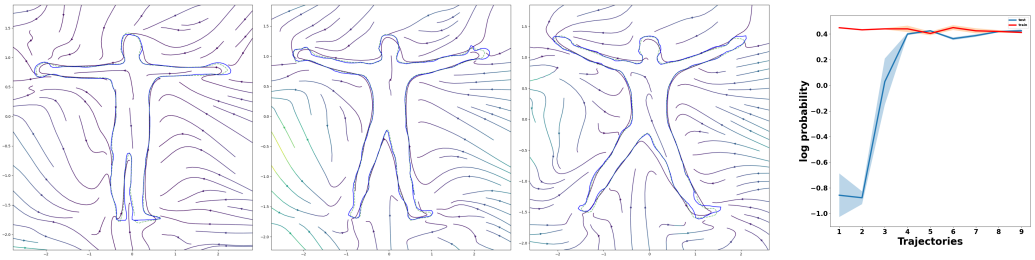

Figure 1: Left, learned stable dynamics for context(0.0, 0.5 and 1.); in blue, the real trajectory; in green, the generated one. The vector field represents the learned dynamic system. The middle case was not part of the training set, but properly generalized to the context. Right, obtained log-likelihoods for train and test data with different amount of training trajectories.

4.1 Conditioned Vitruvian man

We have considered the problem of learning a highly nonlinear 2D limit cycles with a low dimensional context state. We have draw 10 trajectories of a man with his arms and legs in different positions, shown in Fig 1. The context state $\boldsymbol{c}$ is a phase parameter that takes values between 0 and 1.

We evaluate in this experiment the capability of our structured dynamics to generalize to both unseen position states and context state. As shown in Fig 1, the policy was able to impose stable motions towards our demonstrated trajectories in all the position state-space.

To evaluate the generalization with different contexts, and given we can compute the exact log-likelihood, we computed the log-likelihood of some test trajectories in regions of the context-state that was not considered in the training. As shown, in Fig. 1, the policy was able to extrapolate the behavior with as few as 5 trajectories.

4.2 Obstacle Avoidance with Stable Policies

Through this experiment, we wanted to validate our structured policies for learning go-to motions instead of limit cycles. We also wanted to study the capacity of our policy to contextualize in higher dimensional contexts. We considered the problem of learning an obstacle avoidance policy while being attracted to the target. We considered 3 obstacles position and the target position as the context information and learned the dynamics by Maximun Likelihood Estimation (MLE) on the expert demonstrations.

In Fig. 2, we show the qualitative results of the learned dynamics. The model was able to learn obstacle avoidance dynamics from expert demonstrations given arbitrary positions of the obstacles and while remaining stable towards the target goal. Figure 2 also shows the applied diffeomorphic transformation in the state-space. Given a rectangular grid in the latent space, the INN learns a deformation of the space around the obstacle to avoid the collisions.

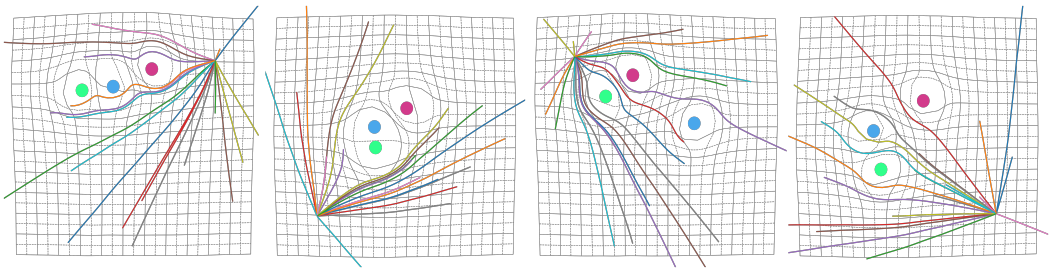

Figure 2: Learned obstacle avoidance trajectories. The deformation in the grid space represents the learned conditioned diffeomorphic mapping with the INN. The latent linear stable dynamics are deformed to avoid the obstacles. Each colored line is a generated trajectory from a random starting pose, while the colored circles are the obstacles conditioning the INN.

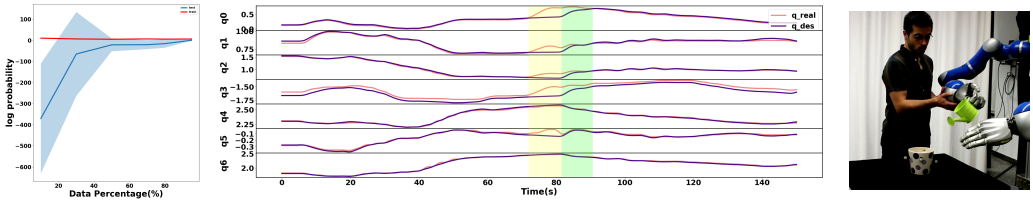

Figure 3: Left, the obtained log probability's mean and standard deviation in the train and test data when training with different percentages of the total data. Center, recorded robot state(pink) and robot commands(purple) during pouring task. In yellow block, human disturb the robot modifying its position. In green block, our policy react to the force, generating new commands. Right, learned pouring task.

4.3 Robot Pouring

In this experiment, we evaluate the proposed dynamics as a policy in a real robot pouring task. In this experiment, we evaluate the scalability of the proposed stable dynamics model to higher dimension dynamics. We also evaluate the robustness of the policy in front of external perturbances and context modifications. Our state is defined by the robot's joint state. The context state is the 6D(position and orientation) of the target pot. We recorded via kinesthetic teaching 220 trajectories of the desired skill with different contexts. The robot should learn the motion to water the pot without dropping the water in the trajectory towards the pot. We performed MLE in the demonstrations and as shown in Fig 3(left), using the whole training data, we obtained test performances on the test data that are in line with the training ones.

We evaluate the performance of the learned dynamics in a real scenario, in which the robot's motion was perturbed both by human external forces and by the modification of the context state. In Fig. 3, the center, we show the case of adding human disturbances. In the yellow region, a human performs some external force in the robot, moving it away from the policy provided trajectory. Then the policy adapts to the disturbance generating new motion commands.

5 Related Work

Stable Dynamics There have been several works trying to learn stable policies. One of the first approaches of this family is Stable Estimator of Dynamical Systems (SEDS) by Khansari-Zadeh and Billard [2011], which is the foundation of many other techniques Khansari-Zadeh and Billard [2014], Neumann and Steil [2015], Ravichandar et al. [2017]. Similar in spirit to our work are Neumann and Steil [2015] and Perrin and Schlehuber-Caissier [2016]. In Neumann and Steil [2015], τ-SEDS is proposed. They combined a quadratic invertible transformation with SEDS, increasing the representation capacity. In Perrin and Schlehuber-Caissier [2016], the proposed invertible transformation was a locally weighted nonparametric translation. In our work, we considered a much richer family of invertible transformation such as INN and extended the family of solutions to Stochastic Dynamic Systems, similarly to Urain et al. [2020], Rana et al. [2020].

Conditioned Dynamics Related with the context state $\boldsymbol{c}$, most of the previous works consider simple affine transformations given the context, (Schaal [2006], Calinon [2016], Khansari-Zadeh and Billard [2014]) and the context information must be closely related with the desired dynamics(goal target position). In our approach, given Conditioned INN are used as the bijective transformation, we can condition our dynamics in arbitrary contexts $\boldsymbol{c}$ and apply highly nonlinear morphing on our dynamics, given the context.

6 Conclusions

We have presented a new policy representation, that can be conditioned on arbitrary context with highly nonlinear transformation while remaining globally asymptotically stable. We have presented our results in a set of Behavioural Cloning situations to prove the policy's representation power.

References

Stefan Schaal. Learning from demonstration. In *Advances in Neural Information Processing Systems*, pages 1040–1046, 1997.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21 (4):682–697, 2008.

Julen Urain, Michele Ginesi, Davide Tateo, and Jan Peters. Imitationflows: Learning deep stable stochastic dynamic systems by normalizing flows. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

N. Hogan. Impedance control: An approach to manipulation. In *1984 American Control Conference*, pages 304–313, 1984.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning(ICML)*, volume 37, pages 1530–1538, 2015.

S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9(1):1–29, 2016.

Vikas Sindhwani, Stephen Tu, and Mohi Khansari. Learning contracting vector fields for stable imitation learning. *arXiv preprint arXiv:1804.04878*, 2018.

S Mohammad Khansari-Zadeh and Aude Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

Klaus Neumann and Jochen J Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70:1–15, 2015.

Harish Chaandar Ravichandar, Iman Salehi, and Ashwin P Dani. Learning partially contracting dynamical systems from demonstrations. In *Conference on Robot Learning*, pages 369–378, 2017.

Nicolas Perrin and Philipp Schlehuber-Caissier. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems & Control Letters*, 96:51–59, 2016.

Muhammad Asif Rana, Anqi Li, Dieter Fox, Byron Boots, Fabio Ramos, and Nathan Ratliff. Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems. *arXiv preprint arXiv:2005.13143*, 2020.

Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

A Stochastic Stable Dynamics by Invertible Neural Networks

In this section, we introduce the stability proof for the model presented in (6). Remark, that the stability guarantees are studied in the continuous time domain, while (6) represents a discretised version of the dynamics. We can easily obtain the discretised model by Euler-Maruyama method.

Model We study the stability guarantees of a class of dynamic systems, composed by a latent($\boldsymbol{Z}$) stable stochastic dynamic system and a deterministic, invertible transformation ($f : \mathbb{R}^d \to \mathbb{R}^d$) between latent space ($\boldsymbol{Z}$) and the observation space ($\boldsymbol{Y}$)

$$\begin{aligned} d\boldsymbol{z}(t) &= v(\boldsymbol{z})dt + \boldsymbol{g}(\boldsymbol{z})d\boldsymbol{B}(t) \\ \boldsymbol{y} &= f(\boldsymbol{z}), \end{aligned} \tag{9}$$

where $v : \mathbb{R}^d \to \mathbb{R}^d$ is a continuous function, $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a continuous matrix function, and $\boldsymbol{B} : \mathbb{R} \to \mathbb{R}^d$ is a $d-$dimensional *Brownian motion* (also called *Wiener process*). We prove the stability guarantees for the dynamic system in $\boldsymbol{Y}$ space

$$d\boldsymbol{y}(t) = J(\boldsymbol{y})v(f^{-1}(\boldsymbol{y}))dt + J(\boldsymbol{y})\boldsymbol{g}(f^{-1}(\boldsymbol{y}))d\boldsymbol{B}(t) \tag{10}$$

where $J(\cdot)$ is the jacobian matrix $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}}$. For simplicity of the derivations, we rewrite (10) as

$$d\boldsymbol{y}(t) = v_y(\boldsymbol{y})dt + \boldsymbol{g}_y(\boldsymbol{y})d\boldsymbol{B}(t) \tag{11}$$

Stability Theory We study the stability guarantees through the Lyapunov Stability method applied for Stochastic dynamics. Assume there exist a $V(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$, $(\boldsymbol{y}, t) \mapsto V(\boldsymbol{y}, t)$ Lyapunov candidate. The time derivative for $V(\boldsymbol{y}, t)$ can be expressed by Itô's formula

$$\begin{aligned} dV(\boldsymbol{y}, t) &= LV(\boldsymbol{y}, t)dt + V_{\boldsymbol{y}}(\boldsymbol{y}, t)\boldsymbol{g}_y(\boldsymbol{y}, t)d\boldsymbol{B}(t) \\ LV(\boldsymbol{y}, t) &= V_t + V_{\boldsymbol{y}}v_y(\boldsymbol{y}, t) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{g}_y^\intercal(\boldsymbol{y}, t))V_{\boldsymbol{yy}}\boldsymbol{g}_y(\boldsymbol{y}, t), \end{aligned} \tag{12}$$

where $LV(\boldsymbol{y}, t)$ is known as the diffusion equation. From (12), the expected value of $\dot{V}$ is

$$\mathbb{E}\left[\dot{V}(\boldsymbol{y}, t)\right] = LV(\boldsymbol{y}, t). \tag{13}$$

To prove stability of our stochastic dynamic system in $\boldsymbol{Y}$, two conditions must be satisfied. Given strictly increasing functions μ_1, μ_2, μ_3, if

$$\mu_1(|\boldsymbol{y}|) \le V(\boldsymbol{y}, t) \le \mu_2(|\boldsymbol{y}|), \tag{14a}$$

$$LV(\boldsymbol{y}, t) \le -\mu_3(|\boldsymbol{y}|)\, \forall \boldsymbol{y} \in \mathbb{R}^d. \tag{14b}$$

then, our stochastic dynamic system (10) is stochastically asymptotically stable.

Proof We define the following Lyapunov candidate

$$V(\boldsymbol{y}) = U(\boldsymbol{z}) = U(f^{-1}(\boldsymbol{y})), \tag{15}$$

were $U(\cdot)$ is a valid Lyapunov candidate for the latent dynamics in $\boldsymbol{Z}$. Given the condition (14a) is satisfied for $U(\boldsymbol{z})$, from the equality in (15), the condition is also satisfied for $V(\boldsymbol{y})$.

To evaluate condition (14b), the diffusion equation for $V(\boldsymbol{y})$ must be computed

$$LV(\boldsymbol{y}) = V_{\boldsymbol{y}}J(\boldsymbol{y})v(f^{-1}(\boldsymbol{y})) + \frac{1}{2}\mathrm{Tr}((J(\boldsymbol{y})\boldsymbol{g}(f^{-1}(\boldsymbol{y})))^\intercal V_{\boldsymbol{yy}}(J(\boldsymbol{y})\boldsymbol{g}(f^{-1}(\boldsymbol{y})))). \tag{16}$$

We can rewrite $V_{\boldsymbol{y}}$ and $V_{\boldsymbol{yy}}$ in terms of $U_{\boldsymbol{z}}$ and $U_{\boldsymbol{zz}}$

$$V_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{\partial}{\partial \boldsymbol{y}}V(\boldsymbol{y}) = \frac{\partial}{\partial \boldsymbol{z}}U(\boldsymbol{z})\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{y}} = U_{\boldsymbol{z}}(\boldsymbol{z})J^{-1}(\boldsymbol{y}), \tag{17}$$

where $U_{\boldsymbol{z}}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{1\times d}$. Finally, we can rewrite $V_{\boldsymbol{yy}}$

$$\begin{aligned} V_{\boldsymbol{yy}}(\boldsymbol{y}) &= \frac{\partial^2}{\partial \boldsymbol{y}^2}V(y) = \frac{\partial}{\partial \boldsymbol{y}^\intercal}\left(\frac{\partial}{\partial \boldsymbol{y}}V(\boldsymbol{y})\right) = \frac{\partial}{\partial \boldsymbol{y}^\intercal}\left(U_{\boldsymbol{z}}(\boldsymbol{z})J^{-1}(\boldsymbol{y})\right) \\ &= \frac{\partial \boldsymbol{z}^\intercal}{\partial \boldsymbol{y}^\intercal}\frac{\partial}{\partial \boldsymbol{z}^\intercal}\left(U_{\boldsymbol{z}}(\boldsymbol{z})J^{-1}(\boldsymbol{y})\right)J^{-\intercal}(\boldsymbol{y})U_{\boldsymbol{zz}}(\boldsymbol{z})J^{-1}(\boldsymbol{y}). \end{aligned} \tag{18}$$

Rewriting (16) as

$$LV(\boldsymbol{y}) = U_{\boldsymbol{z}}f(\boldsymbol{z}) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{g}^\intercal(\boldsymbol{z})U_{\boldsymbol{zz}}\boldsymbol{g}(\boldsymbol{z})) = LU(\boldsymbol{z}). \tag{19}$$

By hypothesis $LU(\boldsymbol{z})$ satisfies the condition in (14b), therefore the condition is also satisfied by $LV(\boldsymbol{y})$.

The proof above shows that the dynamic system in (6) model is globally stochastically asymptotically stable as long as the latent dynamics are globally asymptotically stable. In the following, we extend the stability proof for Conditioned dynamics.

Conditioned Model We study the stability guarantees for a class of dynamic systems, composed by a latent($\boldsymbol{Z}$) stable stochastic dynamic systems and a **context-based**, deterministic, invertible transformation(f : $\mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^d$), that maps $\boldsymbol{z} \in \mathbb{R}^d$, $\boldsymbol{c} \in \mathbb{R}^n$ to $\boldsymbol{y} \in \mathbb{R}^d$

$$\begin{aligned} d\boldsymbol{z}(t) &= v(\boldsymbol{z})dt + \boldsymbol{g}(\boldsymbol{z})d\boldsymbol{B}(t) \\ \boldsymbol{y} &= f(\boldsymbol{z}, \boldsymbol{c}), \end{aligned} \tag{20}$$

where $\boldsymbol{c}$ is the context variable, and f is bijective between $\boldsymbol{y}$ and $\boldsymbol{z}$.

Context Based Stability For all $\boldsymbol{c} \in \boldsymbol{C}$, the proposed context-based transformation f is bijective and invertible. Therefore, $\forall\, \boldsymbol{c} \in \boldsymbol{C}$ the conditioned model is a stable system.

However, even if for each $\boldsymbol{c} \in \boldsymbol{C}$ the model is stable, this property does not ensure that the Conditioned model will be stable. In order to ensure stability, we must consider additional assumptions on the context dynamics. Assume that the context variable evolves as follows

$$\dot{\boldsymbol{c}}(t) = w(\boldsymbol{y}(t), \boldsymbol{c}(t)) \tag{21}$$

with $||\boldsymbol{c}(t) - \boldsymbol{c}^*|| < \delta$ for all $t > t_0$ and $\boldsymbol{c}(t) \to \boldsymbol{c}^*$ as $t \to \infty$ i.e, the dynamics are asymptotically stable. Thus, as $t \to \infty$, the conditioned model will evolve towards the particular stochastic stable dynamic case

$$\begin{aligned} d\boldsymbol{z}(t) &= v(\boldsymbol{z})dt + \boldsymbol{g}(\boldsymbol{z})d\boldsymbol{B}(t) \\ \boldsymbol{y} &= f(\boldsymbol{z}, \boldsymbol{c}^*). \end{aligned} \tag{22}$$

Thus, the Conditioned model will be globally stochastically asymptotically stable.

A.1 Behavioural Cloning Learning

We can apply MLE for behavioural cloning in our model. Given a set of trajectories $\mathcal{D}_\tau = (\tau_0, \tau_1, \ldots, \tau_K)$, where each trajectory is composed of T context and robot state $\tau_i = (\boldsymbol{y}_0^i, \boldsymbol{c}_0^i, \ldots, \boldsymbol{y}_T^i, \boldsymbol{c}_T^i,)$

$$\theta^* = \arg\max_\theta \log p(\mathcal{D}_\tau; \theta) \tag{23}$$

where,

$$p(\mathcal{D}_\tau; \theta) = \prod_{i=0}^{K} p(\boldsymbol{y}_0^i | \boldsymbol{c}_0^i; \theta) \prod_{t=1}^{T} p(\boldsymbol{y}_t^i | \boldsymbol{y}_{t-1}^i, \boldsymbol{c}_{t-1}^i; \theta). \tag{24}$$

From (8), we can compute the exact probability for $p(\boldsymbol{y}_t^i | \boldsymbol{y}_{t-1}^i, \boldsymbol{c}_{t-1}^i; \theta)$, and so, we can directly compute the MLE.