End-to-End Stable Imitation Learning via Autonomous Neural Dynamic Policies

Dionis Totsila[1†*], Konstantinos Chatzilygeroudis[2†*], Denis Hadjivelichkov[3], Valerio Modugno[3], Ioannis Hatzilygeroudis[1], and Dimitrios Kanoulas[3]

[1]*Computer Engineering and Informatics Department (CEID), University of Patras, Greece*

[2]*CILab, Department of Mathematics, University of Patras, Greece*

[3]*RPL Lab, University College London, United Kingdom* [†] Equal Contribution

State-of-the-art Policy Structures

Neural Network Policies:

✓ are flexible
✓ are generic

✗ do not extrapolate well
✗ do not provide stability guarantees

Dynamical System Policies:

✓ provide stability guarantees
✓ behave well in out-of-distribution regions

✗ are not generic
✗ are not flexible

Autonomous Neural Dynamic Policies:

✓ provide stability guarantees
✓ are robust with out of distribution inputs
✓ are generic
✓ more flexible than traditional DS Policies

Proposed Policy Structure

We make the assumption that the state of the system can be split into two parts: (a) a part that can be directly controlled (e.g., positions and velocities of the end-effector), and (b) a part that can only be observed and/or indirectly controlled (e.g., obstacles/objects). In particular (we omit the time notation, t, for clarity):

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_c \\ \boldsymbol{x}_{nc} \end{bmatrix} \in \mathbb{R}^{d_c + d_{nc}}, \tag{1}$$

where $\boldsymbol{x}_c$ is the part of the state that can be directly controlled and $\boldsymbol{x}_{nc}$ is the part of the state that can only be observed. d_c and d_{nc} are the state-space dimensions for the controllable and non-controllable parts, respectively ($d_c + d_{nc} = E$).

We define the control policy as a dynamical system with a fixed attractor $\boldsymbol{x}_c^*$ (formulated as a weighted sum of elementary linear dynamical systems):

$$\dot{\boldsymbol{x}}_c = \pi(\boldsymbol{x}) = \sum_{i=1}^{N} w_i(\boldsymbol{x}) \boldsymbol{A}_i \Big(\boldsymbol{x}_c^* - \boldsymbol{x}_c \Big) \tag{2}$$

where N is the number of elementary dynamical systems, $w_i(\boldsymbol{x}) \in \mathbb{R}$ are state-dependent weighting functions, and $\boldsymbol{A}_i \in \mathbb{R}^{d_c \times d_c}$, $\boldsymbol{x}_c^* \in \mathbb{R}^{d_c}$.

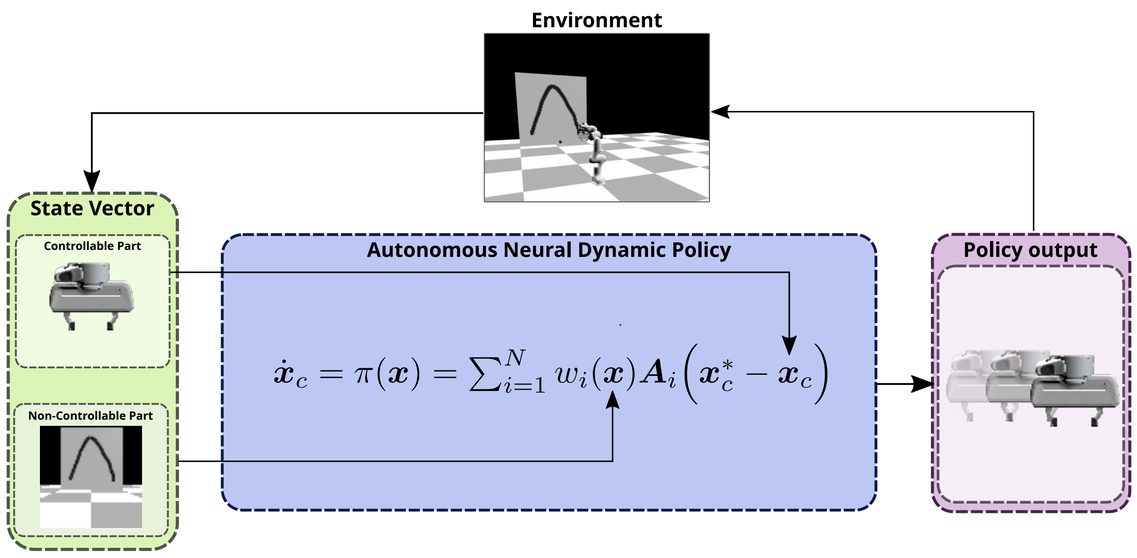

Figure 1. Autonomous Neural Dynamic Policy Outline.

The control policy, $\pi(\boldsymbol{x})$ (Fig. 1), defines the desired velocity profile that the controllable state $\boldsymbol{x}_c$ should follow. Depending on the state representation one can directly use the output for commanding the robot, use a PD controller, or use some inverse dynamics/kinematics model.

Stability Guarantees

Assume that the controllable part of a state trajectory follows the policy as defined in Eq. 2. Then, the function described by Eq. 2 is asymptotically stable to $\boldsymbol{x}_c^*$ if

$$\begin{cases} \boldsymbol{A}_i + \boldsymbol{A}_i^T \succ 0 & \text{the symmetric part of } A \text{ is psd} \\ w_i(\boldsymbol{x}) > 0, & i = 1, .., N, \forall \boldsymbol{x} \in \mathbb{R}^E \end{cases} \tag{3}$$

The proof follows classical Lyapunov analysis.

Funding

Konstantinos Chatzilygeroudis was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "3rd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers" (Project Acronym: NOSALRO, Project Number: 7541). Dimitrios Kanoulas and Valerio Modugno were supported by the UKRI Future Leaders Fellowship [MR/V025333/1] (RoboHike).

Multi-Task Learning with Image Inputs

1. We collect one demonstration for each of the following movements: **a)** a sinusoidal motion, **b)** a linear motion, and **c)** a curvilinear motion.
2. We use the non controllable part of the state to "define" which task we want the robot to perform.
3. In order to create the labels, we simulate a sign with the image corresponding to every motion.

Evaluation

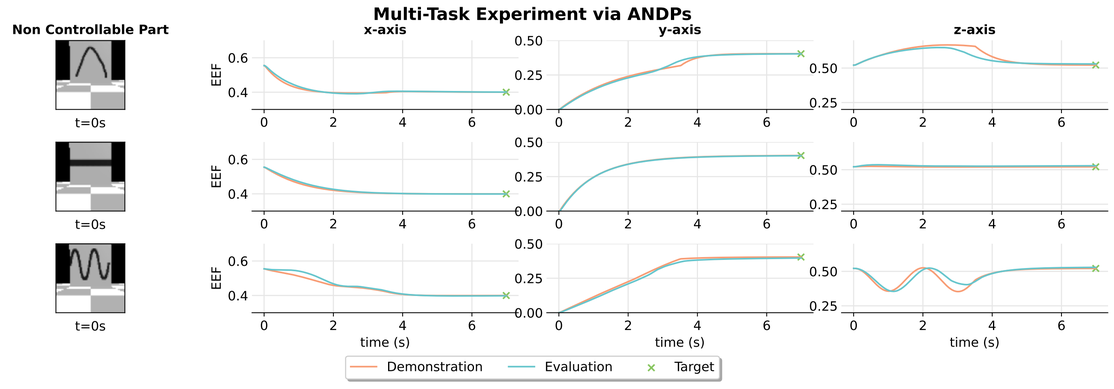

Figure 2. Multi-task scenario with image inputs. All tasks are learned with a single model that can distinguish between tasks given an image input.

ANDPs are reactive to state changes!

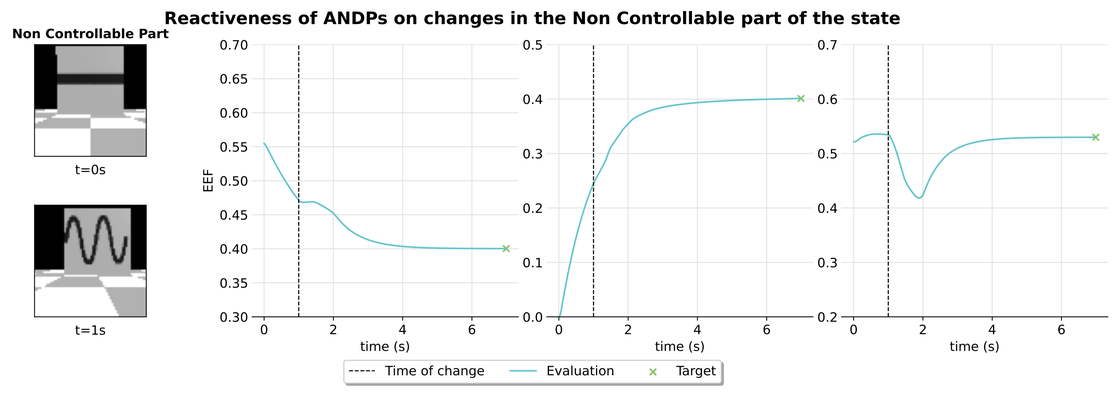

Figure 3. Reactiveness of ANDPs on changes in the non-controllable part of the state, we switch from the line image to the sine image at t=1s.

ANDPs are robust to external force perturbations!

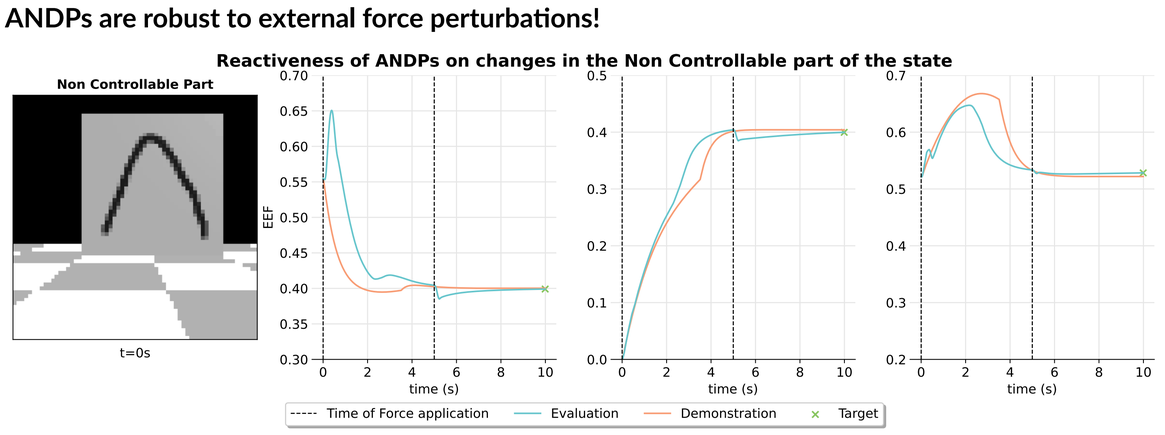

Figure 4. Reactiveness of ANDPs on external force perturbations we apply an external force twice, one at $t = 0s$ and $t = 5s$.

Physical Robot Experiment

Figure 5. Collecting pouring task demonstrations via kinesthetic guidance on the Franka Emika Panda robot.

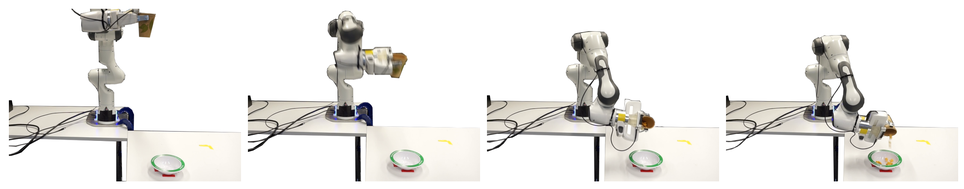

Figure 6. From left to right, screenshots of a successful trial of the pouring task in the physical setting.

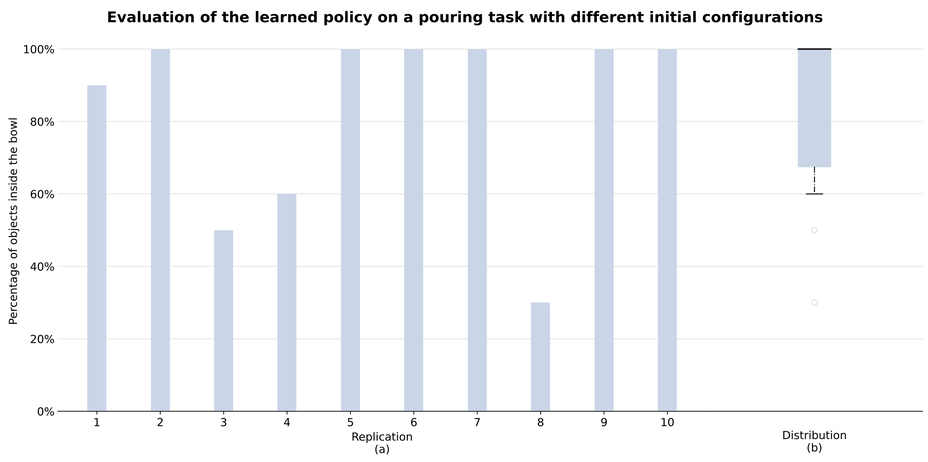

Figure 7. Percentage of objects that ended up inside the bowl on the 10 replications of the pouring task.

*Corresponding authors: dtotsila@upnet.gr costashatz@upatras.gr