Bio-inspired Adaptive Control of Robotic Manipulators for Space Debris Removal and On-Orbit Servicing

ASTRA 2023

17th Symposium on Advanced Space Technologies in Robotics and Automation

Collins Ogundipe[a], Alex Ellery[a]

[a] *Centre for Self-Replication Research (CESER)*
Department of Mechanical & Aerospace Engineering
Carleton University, Canada
collinsogundipe@cmail.carleton.ca
aellery@mae.carleton.ca

18 - 20 October 2023

Content Outline

Introduction
Space Manipulators for Debris Removal

The operations may be divided into 3 stages:

- Stage (I): Free-space manipulator arm motion to aim for grasping the target (**position control**)

Free-flyer's position/base stability control by other mounted manipulators
Reaction Wheels/Thrusters deployed for attitude control

- Stage (II): Transitional contact dynamics and passivation of target to grapple and remove.

For grasping, **hybrid position/force control** and **impedance control** are most popular methods.

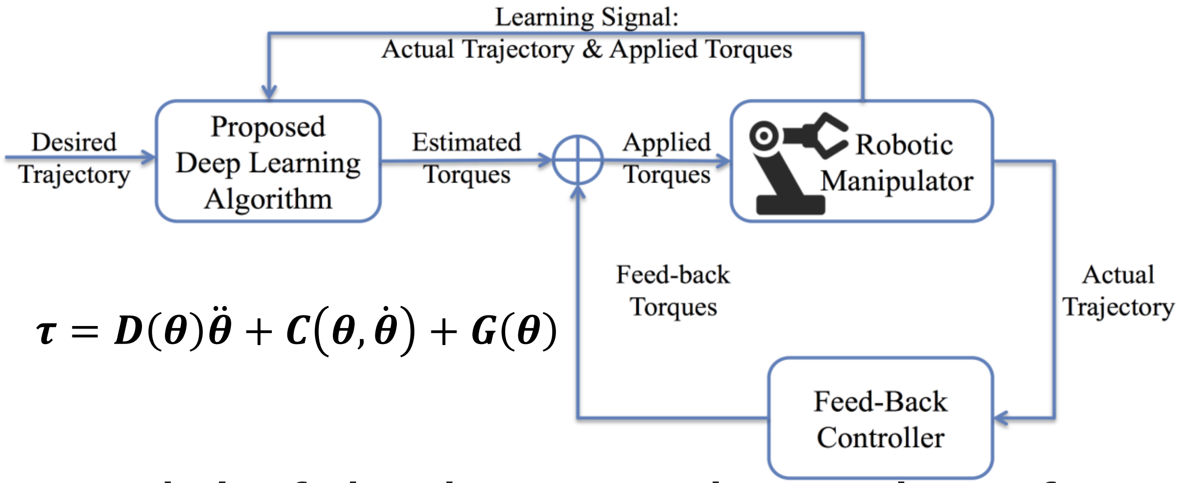
- Stage (III): On-orbit Servicing Operations (e.g., peg-in-hole task)

Hybrid position/force control, as option.
Impedance control (adopted) – hybrid position/force control does not take into consideration the impedance effect between the environment and the robot end effector.

Introduction

Feedback Control Problem in Space Manipulation

- Tracking desired trajectory by integrating the acceleration policy while deploying feedback control to reject the modeling errors of the dynamics



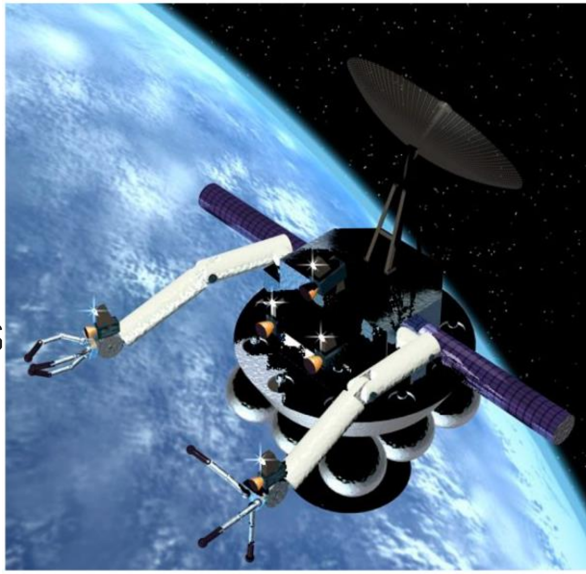
$$\tau = D(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta)$$

- But tracking is only feasible with accurate model of the inverse dynamics of the system

- Trade-off between the compliancy and reactiveness of the controller against accuracy, as effort is being made to tune the gains (higher)

- Machine learning methods have been explored that can learn and improve this inverse dynamics approach

Problem Statement: Why Robotic Manipulation Approach?



Ellery, A. An Introduction to Space Robotics, Praxis– Springer Series on Astronomy and Space Sciences, 2000.

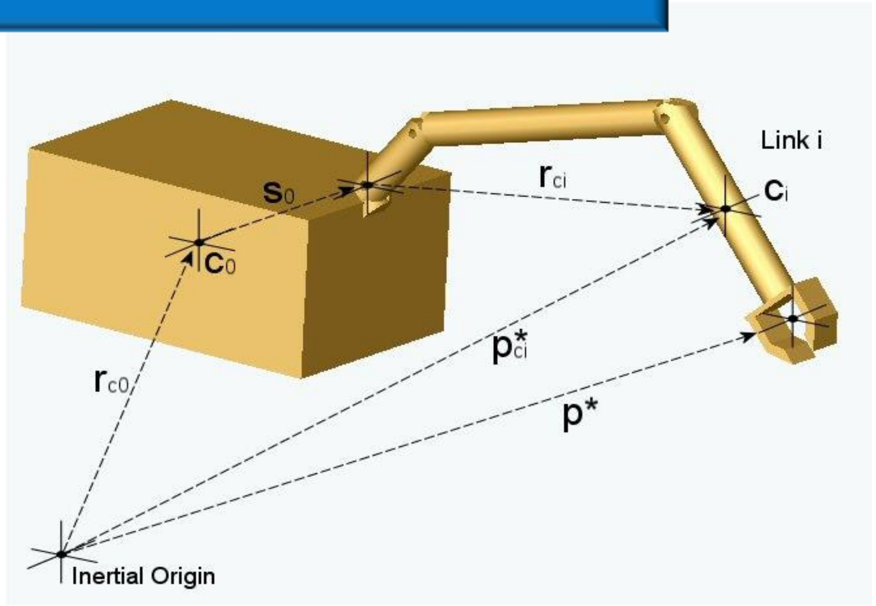
- Only robotic manipulation is flexible enough to deal with both large and small debris
- Harpoons and nets generate complex uncontrollable dynamic interactions
- Free-flyer spacecraft mounted with dexterous manipulators will provide controlled interaction with target
- Robotic manipulator offers possibility for re-use and on-orbit servicing
- Possible to exhibit human-like tactility in robotic grasping of space debris?
- Force control modeling to achieve adaptable and compliant behavior - partially known payload dynamics at times

The Problem of Space Manipulators
Kinematics of Space-based Robot

- The final equation of space robot's position kinematics

$$p^* = p^*_{cm} + \frac{m_0}{m_T} s_0 + \sum_{i=1}^{n} R_i \lambda_i - \frac{m_{n+1}}{m_T} R_{n+1} r_{n+1}$$

$$\text{where } \lambda_i = \frac{1}{m_T} \sum_{j=0}^{i} (m_j l_j - m_i r_i)$$

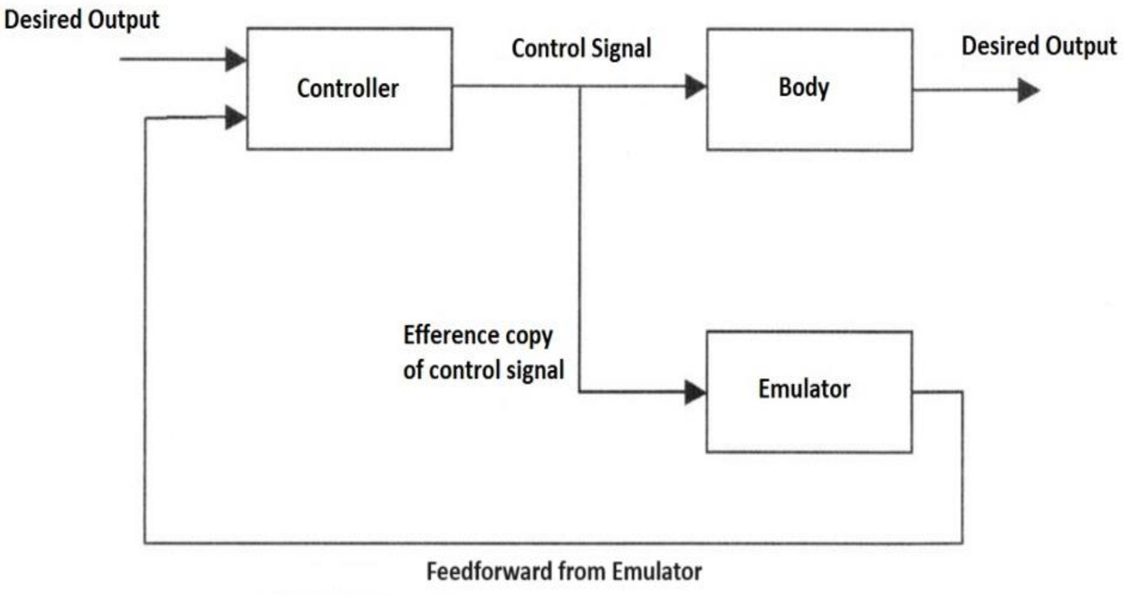


- The substituted terms (pre-calculated direct substitution)

- Space-based environmental factors/forces are negligible

- Fundamentals of transfer learning suggest it should generalize efficiently from earth to space via pre-trained networks, both with similar multi-dimensional parabolic and kinematics.

Bio-Inspired Predictive Feedforward
Brain/Human Level Manipulation



- An efference copy of the motor control signal is typically transmitted to an emulator (input-output)
- The efference copy of the motor commands then produces a feedforward error compensation
- Biomimetically, a categorized neural network system in any control architecture *can* imitate this function of the motor cortex
- There is a time delay of 40-60ms feeding back the error between the actual motor outputs and the commanded motor input
- Explore predictive neural networks as forward model by adopting input-output models

Bio-Inspired Predictive Feedforward

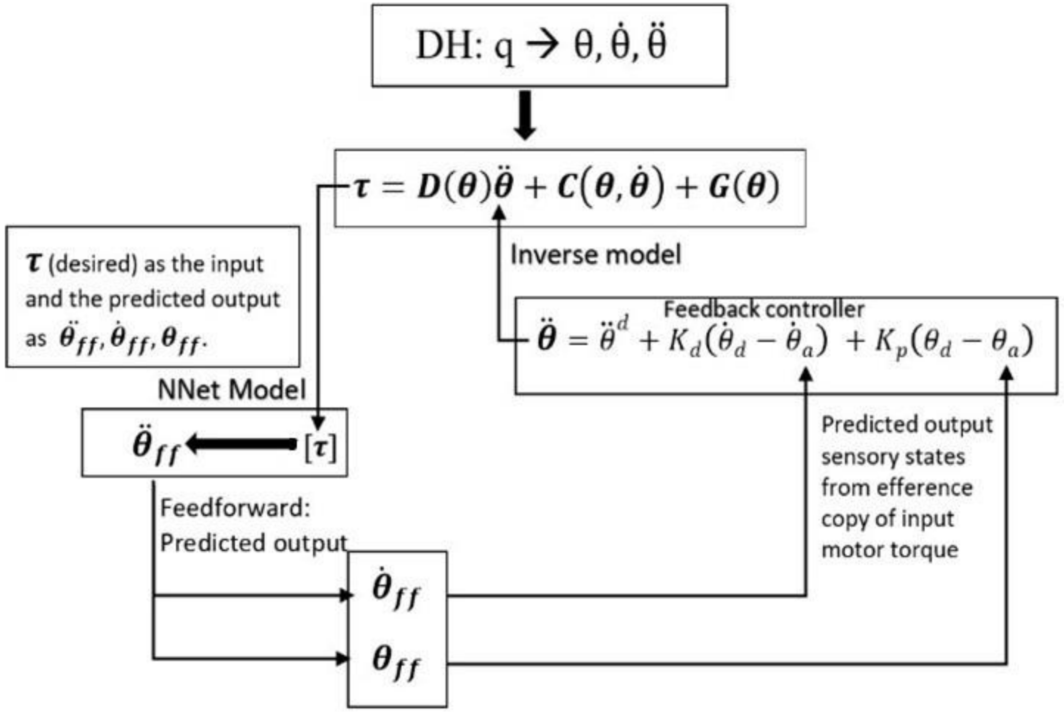
NeuralNet Feedforward Approach

- Traditional control (with feedback signals from sensors) has delays which generate instabilities (and high gains)

- We developed at first neural network models capable of predicting forward trajectory variables $\left(\theta_{ff},\ \dot{\theta}_{ff},\ \ddot{\theta}_{ff}\right)$ from efference (desired) torque
  $\ddot{\theta} = D^{-1}(\theta)\left[\tau - C\left(\theta,\dot{\theta}\right) - G(\theta)\right]$

- Multiple Output Regression Algorithms used where such performed better

- Models poised to cancel the sensory effects of the arm movement, providing anticipated sensory consequences



- Instabilities that could arise in delays from traditional feedback cycle has been partially circumvented

- This is akin to how the human cerebellum functions

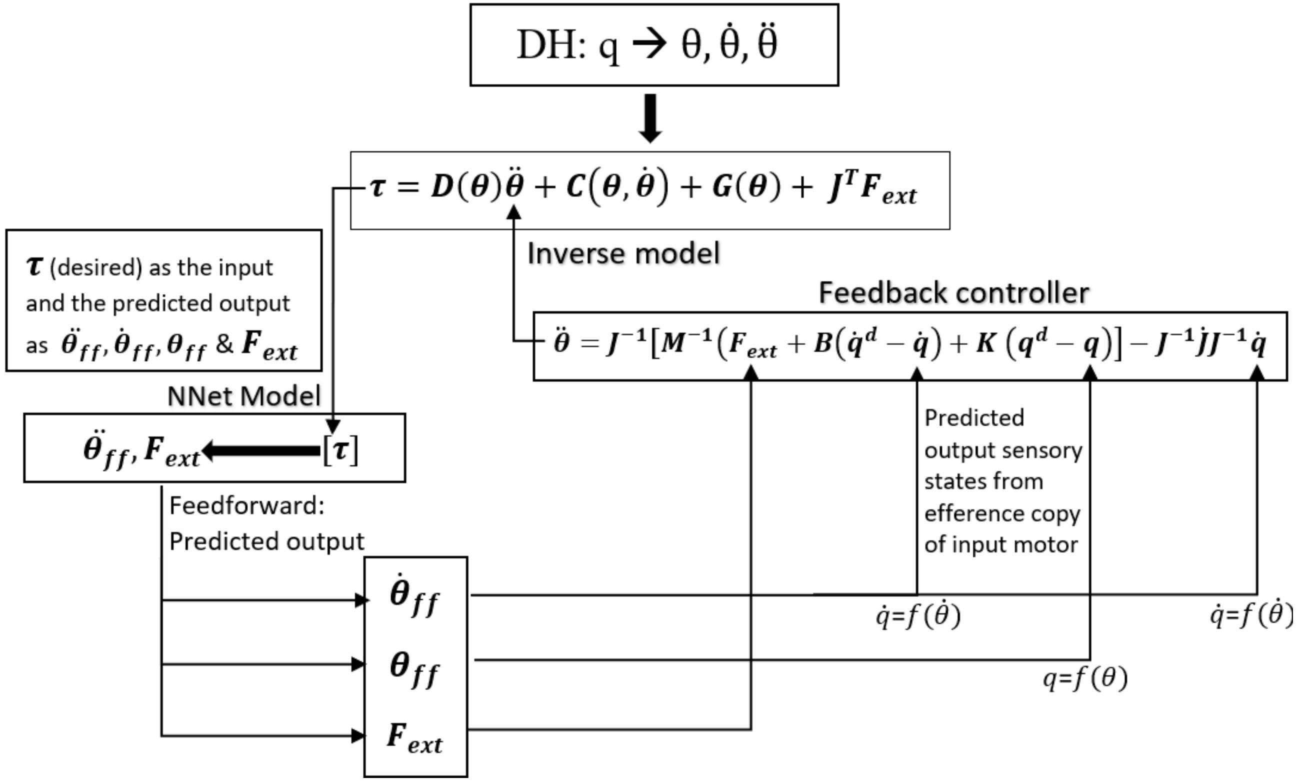
Bio-Inspired Predictive Feedforward
Force Control Model

As impedance control relies on sensory feedback which are subject to time delays, instabilities can quickly arise when controlling forces.

Forward models are therefore crucial to compensating for this:

$$\ddot{\theta} = D(\theta)^{-1}\left[\tau - C\left(\theta, \dot{\theta}\right) - G(\theta) - J^T F_{ext}\right]$$

The impedance controller could operate independently of the forward model to change the impedance (stiffness) of the limb/arm joints.



$$\ddot{q} = M^{-1}\left(F_{ext} + B\left(\dot{q}^d - \dot{q}\right) + K\left(q^d - q\right)\right)$$

Joint space	Cartesian space
$\dot{\theta} = J(\theta)^{-1}\dot{q}$	$\dot{q} = J(\theta)\dot{\theta}$
$\ddot{\theta} = J(\theta)^{-1}\ddot{q} - J(\theta)^{-1}\dot{J}(\theta)J(\theta)^{-1}\dot{q}$	$\ddot{q} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta}$

Bio-Inspired Predictive Feedforward

Feedforward Training Scheme

Dataset:

7 d.o.f WAM Barrett Arm datasets – 12,000 samples for learning/training

Multiple Targets Prediction Layout:

For a feature vector *x*, we aim to predict a vector of responses *y* using a function *h(x):*

$$\boldsymbol{x} = (x_1, x_2, \dots, x_p) \overset{h(x)}{\Longrightarrow} \boldsymbol{y} = (y_1, y_2, \dots, y_m)$$

- Training data: $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \dots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, $\boldsymbol{y}_i \in \mathcal{Y} = \mathbb{R}^m$
- Predict the vector $\boldsymbol{y} = (y_1, y_2, \dots, y_m)$ for a given $\boldsymbol{x}$.

	X_1	X_2	Y_1	Y_2	$\dots$	Y_m
$\boldsymbol{x}_1$	5.0	4.5	14	0.3		9
$\boldsymbol{x}_2$	2.0	2.5	15	1.1		4.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\boldsymbol{x}_n$	3.0	3.5	19	0.9		2
$\boldsymbol{x}$	4.0	2.5	?	?		?

Algorithm Challenges:

Appropriate modeling of target dependencies between targets $y_1, y_2, \dots, y_m$, and a multitude of multivariate loss functions defined over the output vector, $\mathcal{L}(y, h(x))$

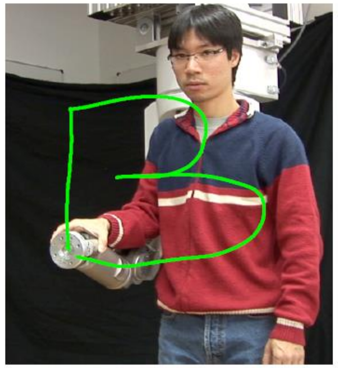
Predictive Feedforward Learning:

Data source: Joint trajectory $(\theta, \dot{\theta}, \ddot{\theta})$ were sampled from the robot and corresponding motor torques (τ) measured for each data point – in teaching mode

Algorithms Deployed:

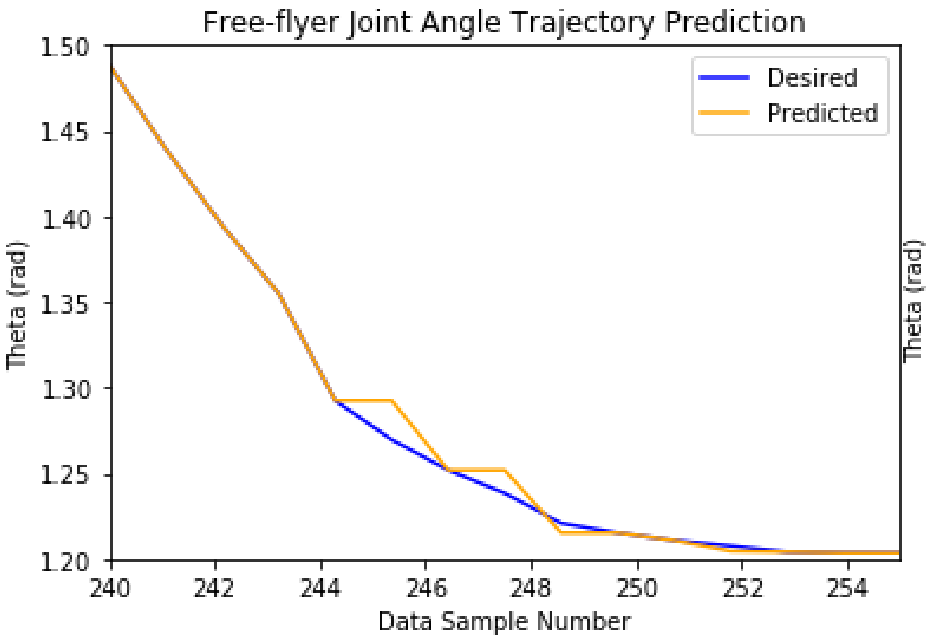
Deep learning multiple-target prediction

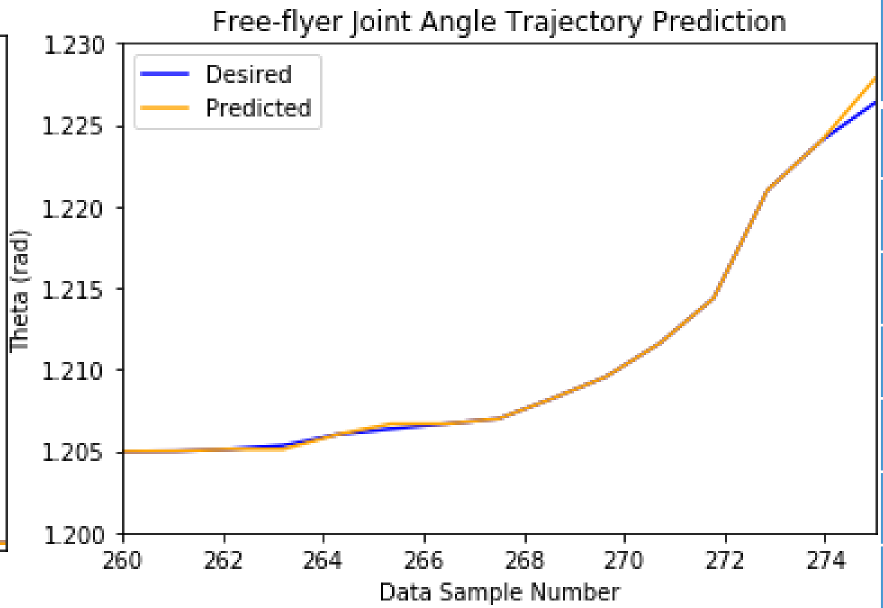
Multiple-output decision tree regression



D. Nguyen-Tuong, M. Seeger, and J. R. Peters. Model Learning with Local Gaussian Process Regression, Advanced Robotics 23 2015–2034, 2009.

The Predictive Feedforward Results





Data 243 Joint Number	Joint Angle (rad) Test Set	Joint Angle (rad) Predicted	Accuracy (%)
1	0.0814	0.0799	98.2
2	0.6165	0.5696	92.4
3	0.0236	0.0222	94.1
4	1.754	1.6647	95.4
5	0.2123	0.1990	93.7
6	0.0781	0.0751	96.2
7	0.0869	0.0844	97.1

Data 243 Joint Number	Joint Velocity (rad/s) Test Set	Joint Velocity (rad/s) Predicted	Accuracy (%)
1	0.0657	0.0647	98.5
2	-0.1850	-0.1835	99.2
3	-0.1794	-0.1600	89.2
4	-0.0678	-0.0642	95.3
5	-0.0235	-0.0231	98.1
6	0.0478	0.0434	90.8
7	-0.0895	-0.0872	97.4

Trajectory training dataset randomly split (70-80% Training set)

Trained models evaluated across different set of dataset/trajectory to verify consistency of performance

Separate models performed better/best for distinct trajectory parameters $(\theta, \dot{\theta}, \ddot{\theta})$

Models could be built to account for different scenarios, with the availability of more teaching mode datasets – both for free-space and payload modes

Practical Limitations Encountered – Transfer Learning

- Given the earth/space kinematics are of the same form but with only changes in parameters, the two polynomial curve shapes are similar…

$$p^* = p_{cm}^* + \frac{m_0}{m_T} s_0 + \sum_{i=1}^{n} R_i \lambda_i - \frac{m_{n+1}}{m_T} R_{n+1} r_{n+1}$$

$$\text{where } \lambda_i = \frac{1}{m_T} \sum_{j=0}^{i} (m_j l_j - m_i r_i)$$

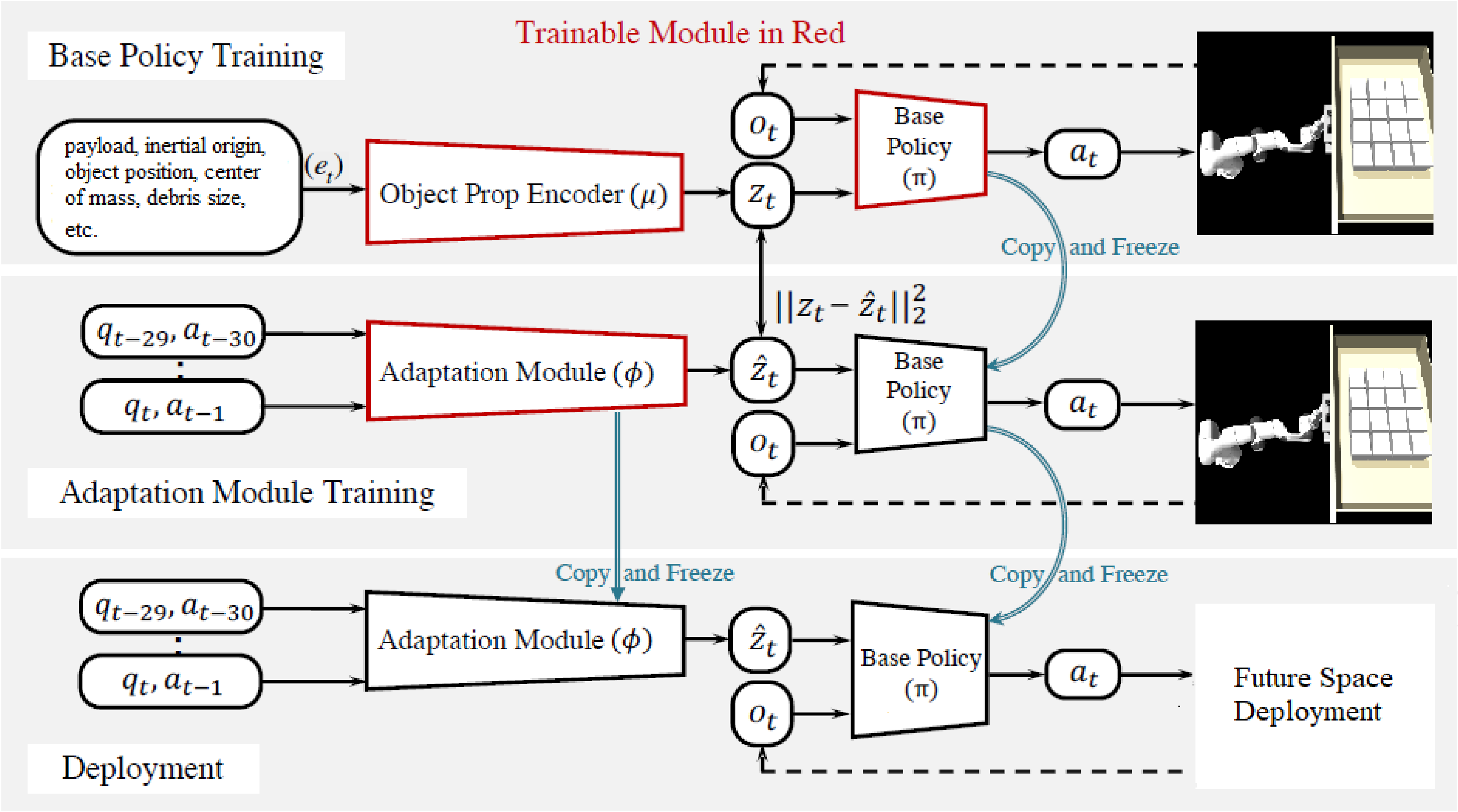
- Transfer learning cannot seem to shift one polynomial curve fit onto the other, but the human cerebellum can

- Different forward model trainings were required for the terrestrial and space robot's joint trajectory predictions to guarantee high accuracy

- Transfer learning lacks the adaptability and requires a large amount of motor models

- There is a need for some offline adaptation or morphing approach between the terrestrial and spaced-based dynamics

Schematic of Approach under RMA



Observation o_t {cur. state}
z_t <-- $\mu(e_t)$ {Prop Enc.}
a_t <-- $\pi(o_t, z_t)$ {action}

Adap module <-> Pred. Fwd Model Err $\{z_t - \hat{z}_t\}$

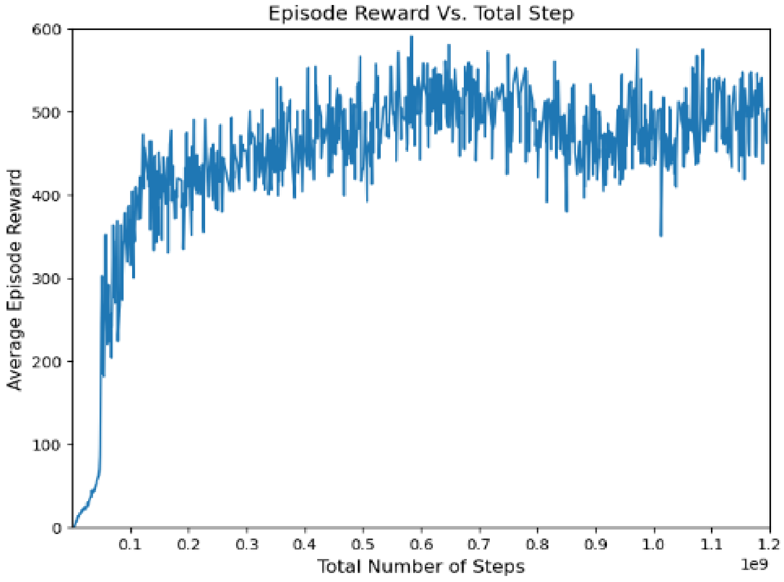
Base Policy Learning – μ and π are jointly optimized using PPO

o_t contains 3 past joint positions and commanded actions.
In **Adaptation**, policy is frozen and SL used to train (ϕ) which uses proprioception and action history (t:t-29; for k=30) to estimate extrinsics vector z_t. During Deployment, the base policy uses $\hat{z}_t$ estimated and updated online by (ϕ)

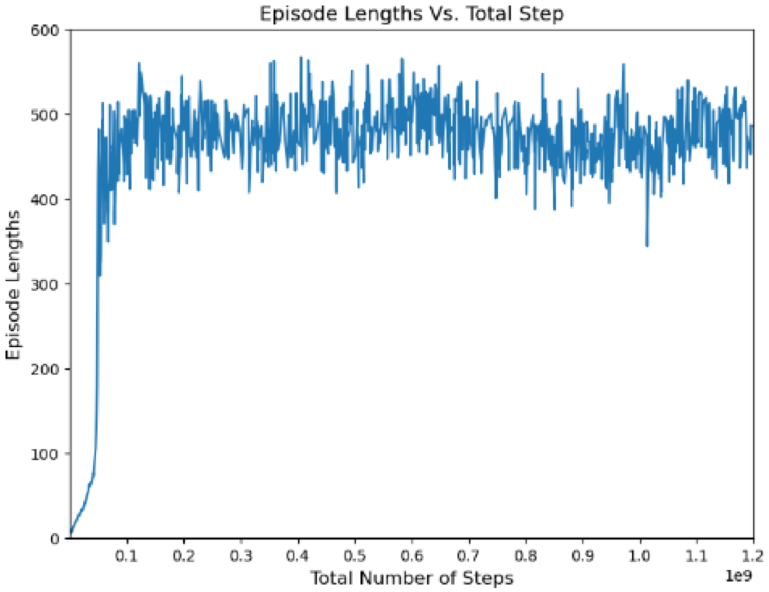
H. Qi, A. Kumar, R. Calandra, Y. Ma, J. Malik. In-Hand Object Rotation via Rapid Motor Adaptation. In Conference on Robot Learning (CoRL), 2022.

The Adaptive Training Results

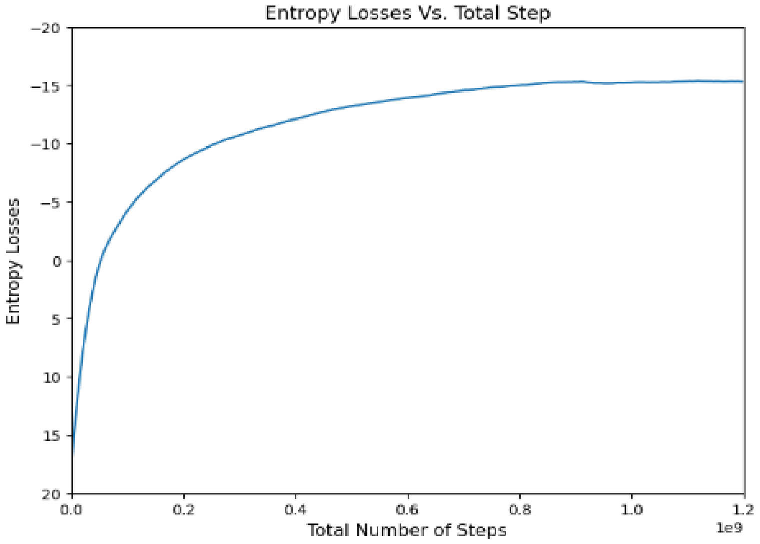
Base Policy (π)



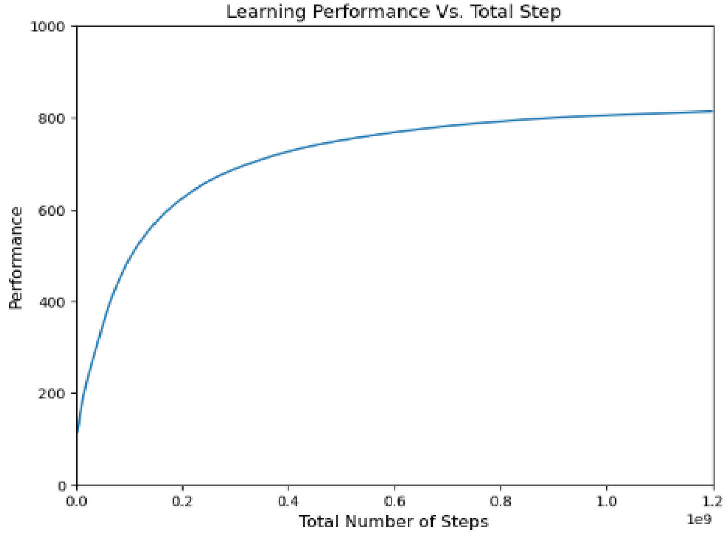
(a)



(b)



(c)



(d)

Results from Base Policy (π) Training on the environment with the implemented RMA algorithms

(a) Average episode reward over training of 1.2 billion steps. Sustained maximized reward shows that policy successfully learned and converged.

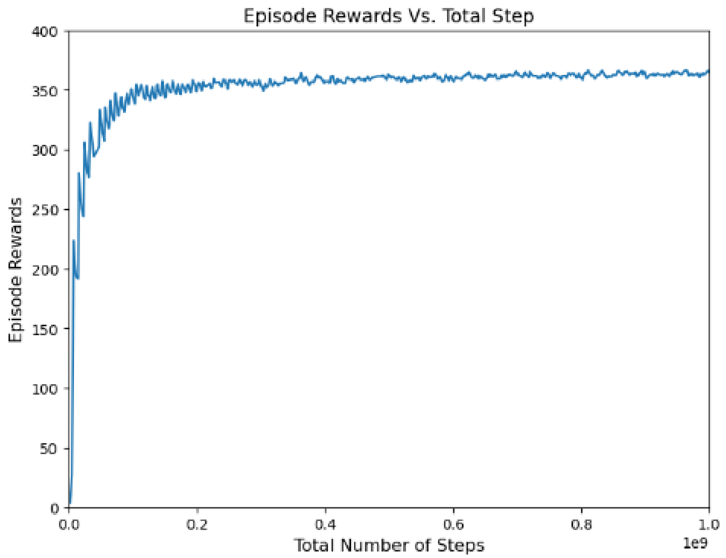
(b) Episode length over training of 1.2 billion steps

(c) The decreasing entropy losses during the total training of 1.2 billion steps

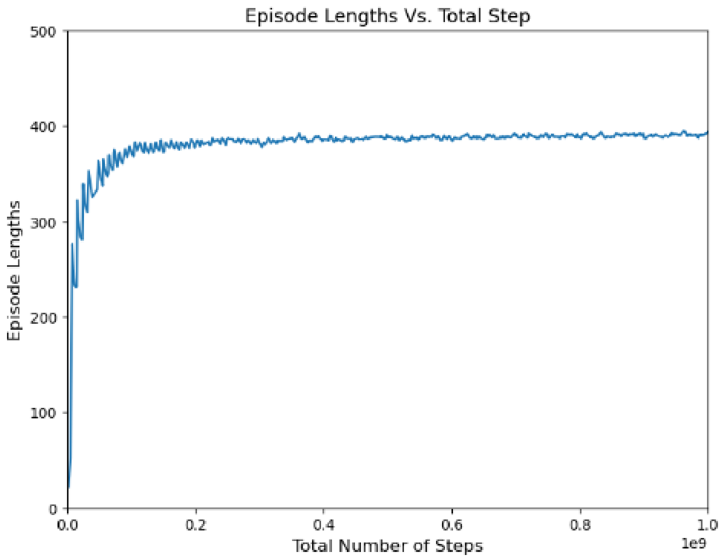
(d) The increasing learning performance during the total training of 1.2 billion steps

Learning performance shows optimal convergence over time

The Adaptive Training Results
Adaptation Module (ϕ)



(a)

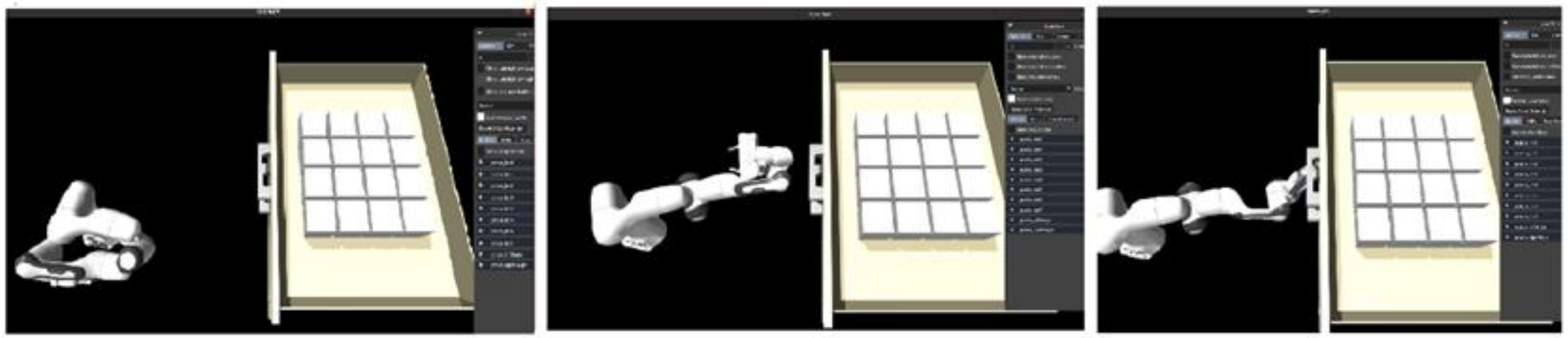


(b)

Results from Adaptation Module (ϕ) Training on the environment with RMA

(a) The average episode reward during total training of 1 billion steps
(b) The episode lengths over 1 billion steps

Reward Function Components

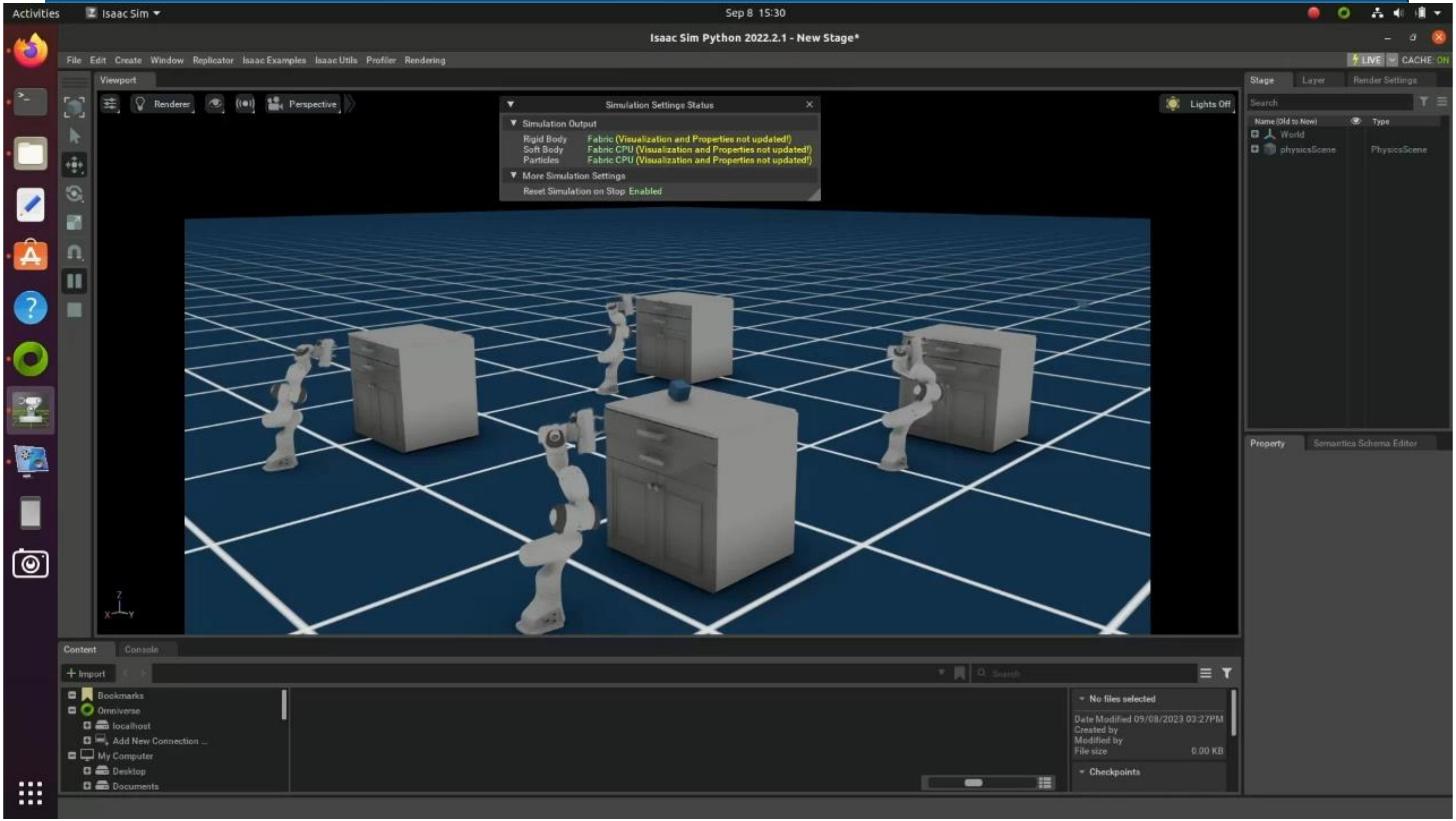


The expected stages of the learning process and rewards associated [NVIDIA IsaacGym]:

Total reward function r to maximize is given by (subscript t omitted):

$$r = 2.0 * dist_{rew} + 0.5 * rot_{rew} + 0.25 * aroundhandle_{reward} + 7.5 * open_{reward} + 5.0 * gripper_{dist\_rew} - 0.01 * action_{penalty}$$

Adaptive Manipulation Results



Conclusions

- Predictive neural-net/regression forward models show promising predictions, with low gains in feedback controller - in simulation
- Developed forward models robust enough to provide a platform for reactive and adaptive robotic manipulation
- There is requirement for some offline adaptation or implementation of morphing approach between the terrestrial and spaced-based dynamics
- Neural nets transfer learning lacks the adaptability of general intelligence
- Rapid motor adaptation via reinforcement learning provides for adaptive and compliant space manipulator control transferable from earth-learned simulation

Thank you!

Department of Mech & Aero.

Carleton University, Canada

collinsogundipe@cmail.carleton.ca

Image credit: ESA/Airbus