ADVANCED ROBOTICS AND MECHATRONICS SYSTEMS LABORATORY

NAZARBAYEV UNIVERSITY

Neural Network based robust control for dual axis reaction wheel inverted pendulum

Yerzhan Rzagaliyev, MSc student
Thesis Supervisor: Prof. Huseyin Atakan Varol
Department of Robotics
Nazarbayev University

Outline

- Motivation
- Physical Setup
- System Dynamics
- System Identification
- Neural Network Implementations
- Simulations
- Experimental Results

Motivation

- Industry 4.0

Internet of Things (IoT)

Big Data

Cloud computing

Augmented and virtual reality (AR/VR)

Human-Robot interaction (HRI)

Additive manufacturing (AM)

Motivation

Variable Impedance/Stiffness Actuators

- For safe HRI used new robots like variable impedance actuators (VIA) (Lasota et al, 2017)
- Highly nonlinear
- Decrease of positioning accuracy
- Vibrations

Reaction wheel integration

- It has benefits to implement reaction wheel-integrated VSA (Baimyshev et al, 2016)
- Can boost the performance of the VSA

Motivation

Reaction wheel inverted pendulum

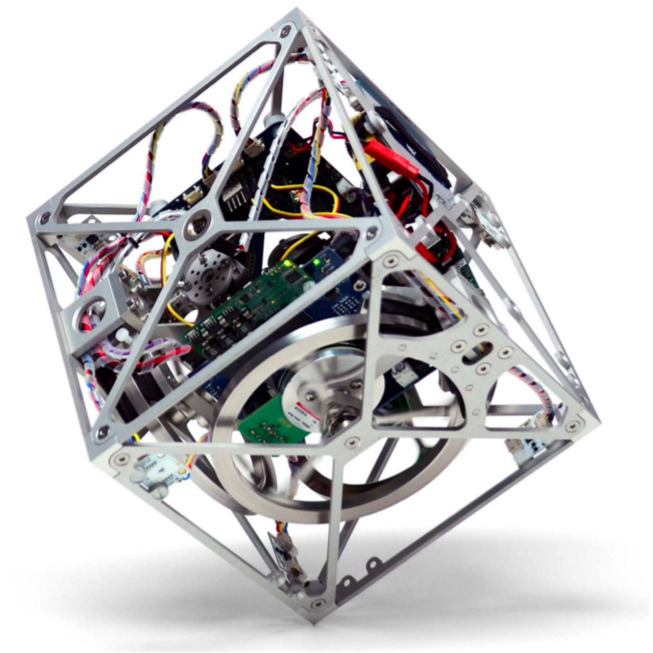


Figure 1. Cubli balancing on a corner (Muehlebach, D'Andrea, 2016)

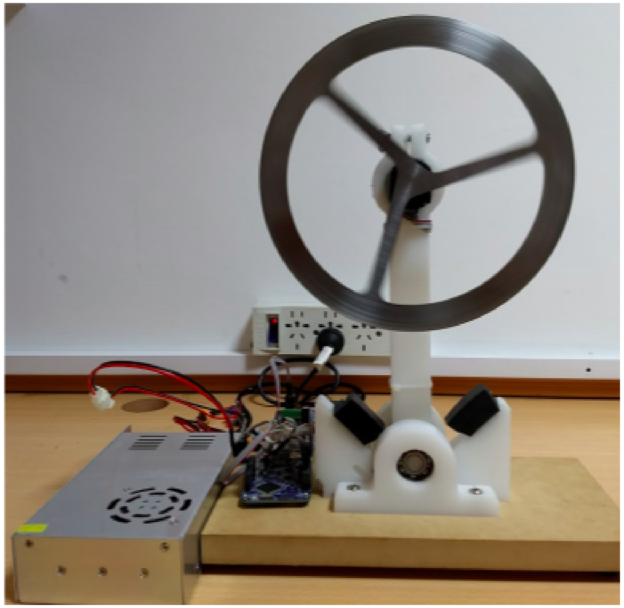


Figure 2. Self balancing single axis inverted pendulum (Belascuen, Aguilar, 2018)

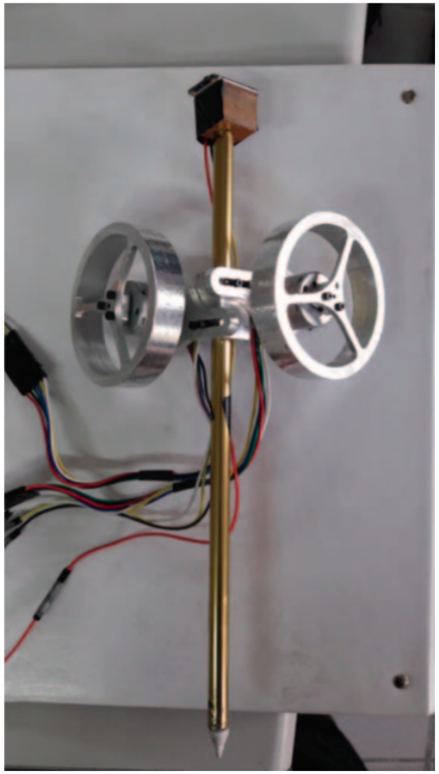


Figure 3. Self balancing dual axis pendulum (Turkmen et. Al, 2017)

Design of the setup

Lower Part

- 3D printed
- 8 springs
- Connected with universal joint
- 2 encoders

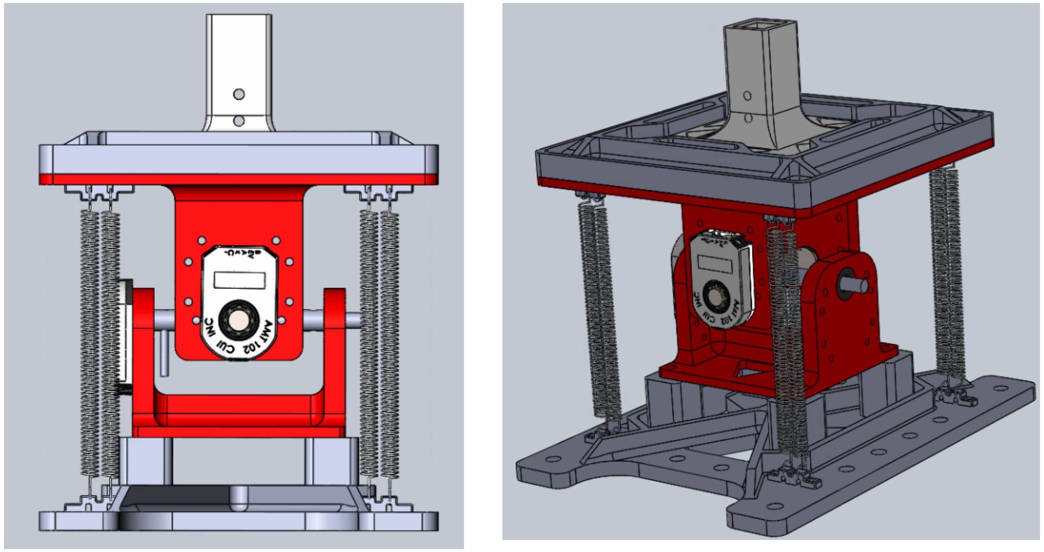


Figure 4. Lower part of the system

Design of the setup

Upper part
- 3D printed

- Steel flywheel
  - Diameter – 8 cm
  - Mass – 160 g

- 2 encoders
- 2 Maxon motors
- 2 Vex gyroscopes

- 20 pieces
  - Mass – 50 g
  - Diameter – 15 mm
  - Thickness – 9 mm

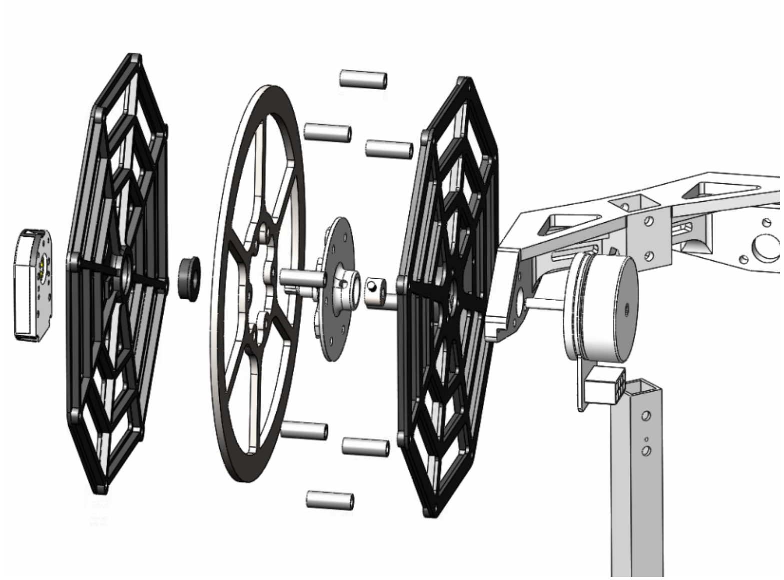


Figure 5. Upper part of the system

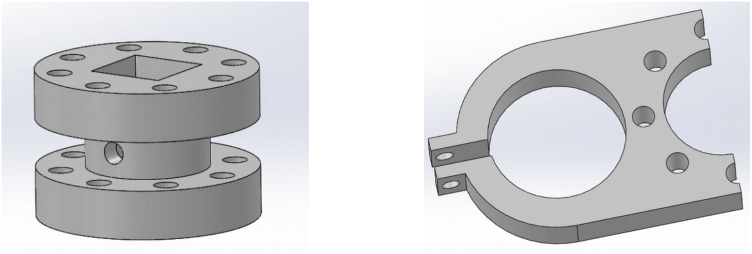
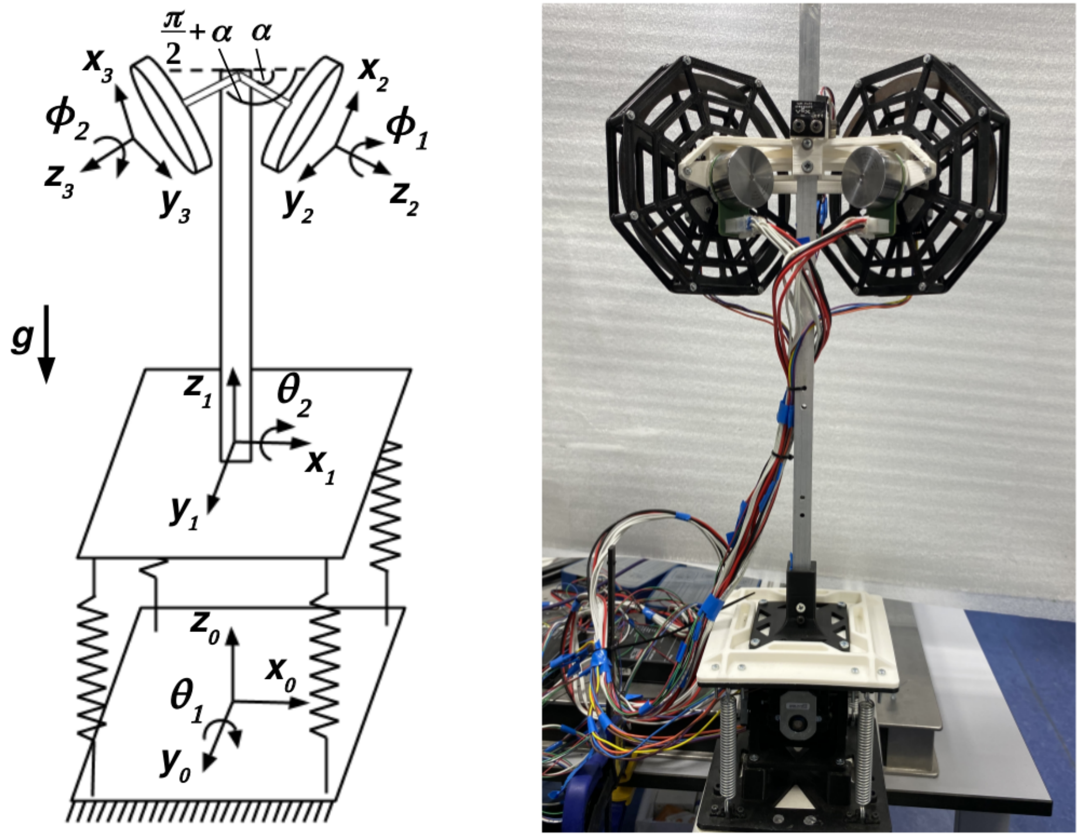


Figure 6. Mass holders

Design of the setup



(a) Block diagram. (b) Reaction wheel setup.

Figure 7. Experimental setup (Baimukashev et al, 2020)

System Dynamics

Kinematics

$$ {}^0_1R = \begin{bmatrix} cos(\theta_1) & 0 & sin(\theta_1) \\ 0 & 1 & 0 \\ -sin(\theta_1) & 0 & cos(\theta_1) \end{bmatrix} $$

$$ {}^1_2R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_2) & sin(\theta_2) \\ 0 & -sin(\theta_2) & cos(\theta_2) \end{bmatrix} $$

$$ \hat{\omega}_p =^0_2 R({}^1_2R^{-1}[0\ \dot{\theta}_1\ 0]^T + [\dot{\theta}_2\ 0\ 0]^T) $$

$$ \hat{\omega}_{\omega_1} =^0_2 RR_{\omega_1}[\dot{\phi}_1\ 0\ 0]^T $$

$$ \hat{\omega}_{\omega_2} =^0_2 RR_{\omega_2}[\dot{\phi}_2\ 0\ 0]^T $$

Lagrangian function

$$ T = \frac{1}{2}\hat{\omega}_p^T I_1 \hat{\omega}_p + \frac{1}{2}\hat{\omega}_{\omega_1}^T I_{\omega_1}\hat{\omega}_{\omega_1} + \frac{1}{2}\hat{\omega}_{\omega_2}^T I_{\omega_2}\hat{\omega}_{\omega_2} $$

$$ V = \bar{g}^T {}^0_2R(m_p l_p + m_{\omega_1} l_{\omega_1} + m_{\omega_2} l_{\omega_2}) + \frac{1}{2}(k_1(\theta_1 - \theta_{1_i})^2 + k_2(\theta_2 - \theta_{2_i})^2) $$

$$ \frac{d}{dt}\frac{\partial}{\partial \dot{q}}L - \frac{\partial}{\partial q}L = \tau $$

$$ \frac{d}{dt}\frac{\partial}{\partial \dot{q}}T - \frac{\partial}{\partial q}T + \frac{\partial}{\partial q}U = \tau $$

$$ M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau $$

$$ \ddot{q} = M(q)^{-1}(\tau - C(q,\dot{q}) - G(q)) $$

System Dynamics

In order to control stabilization of the pendulum, we need to define all affecting forces and torques, and we have equation:

$$\tau = \begin{bmatrix} cos(\theta_2)cos(\alpha)(-K(i_{\omega_1} + i_{\omega_2}) + b_{\omega_1}\dot{\phi}_1 + b_{\omega_2}\dot{\phi}_2) - b_1\theta_1 \\ sin(\alpha)(-K(i_{\omega_1} - i_{\omega_2}) + b_{\omega_1}\dot{\phi}_1 - b_{\omega_2}\dot{\phi}_2) - b_2\theta_1 \\ Ki_{\omega_1} - b_{\omega_1}\dot{\phi}_1 \\ Ki_{\omega_2} - b_{\omega_1}\dot{\phi}_1 \end{bmatrix}$$

By deriving dynamics from Lagrangian function, we have our state-space representation as:

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ (M(q)^{-1}(\tau - C(q, \dot{q}) - G(q)))^T \end{bmatrix}$$

System Identification

In Solidworks

Mass – 1.49 kg
Center of mass (in z axis) – 31 cm
Moments of inertia:

- Ixx = 0.19 kg·m^2
- Iyy = 0.20 kg·m^2
- Izz = 0.01 kg·m^2

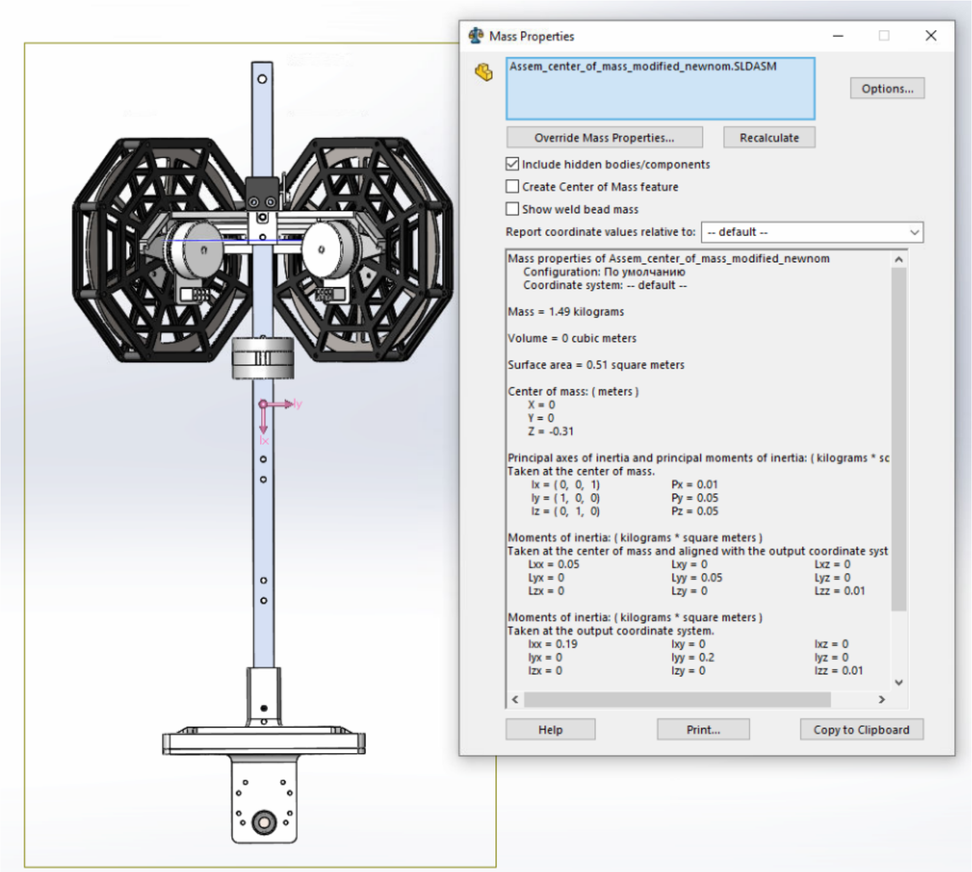


Figure 8. Mechanical properties

System Identification

In MATLAB

Spring constant

- k1 – 12.5 N·m/rad
- k2 – 6.9 N·m/rad

Pendulum friction coefficient

- b1 – 0.09 N·m·s/rad
- b2 – 0.12 N·m·s/rad

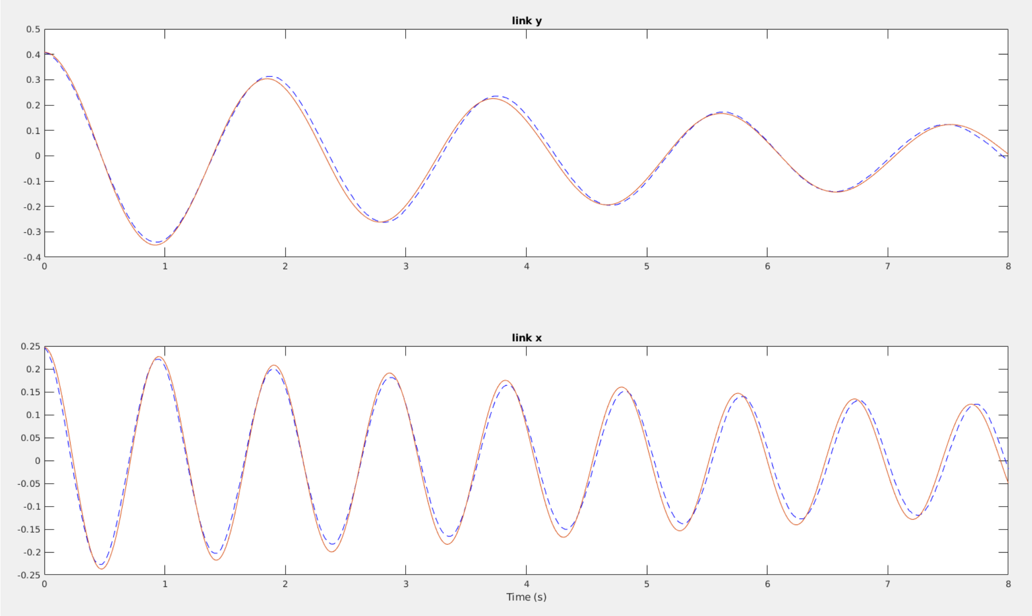


Figure 9. Defining rotational spring constant and pendulum friction coefficient. Real world result (blue dashed line) vs simulated result (red line)

System Identification

Parameter	Symbol	Value	Unit
Mass of the whole pendulum	m_p	1.49	kg
Center of mass along z-axis	l_p	0.31	m
Rotational spring constant around x-axis	k_1	12.5	N ·m/rad
Rotational spring constant around y-axis	k_2	6.9	N ·m/rad
Friction coefficient on x-axis of the pendulum	b_1	0.09	N ·m·s/rad
Friction coefficient on y-axis of the pendulum	b_2	0.12	N ·m·s/rad
Friction coefficient of the reaction wheel	$b_{\omega 1}$	1.11E-04	N ·m·s/rad
Friction coefficient of the reaction wheel	$b_{\omega 2}$	9.40E-05	N ·m·s/rad
Motor torque constant	K	0.13	N ·m/A
Moment of inertia of the pendulum	I_p	0.19	kg·m^2
Moments of inertia of the reaction wheels	$I_1 = I_2$	7.53E-05	kg·m^2

Table 1. Nominal parameters of the system

Neural Network Implementations

Optimal control problem (OCP) trajectories

- MATMPC (Chen et al, 2019)
- 8000 trajectories, after filtering 5300 clean
- 15000 trajectories, after filtering 8000 clean
- Mass – 0 to 30 %, spring constants – -20 to 0 %, friction coefficients – -20 to 20 %
- For testing 500 trajectories for each (nominal and changed)

	Nominal	Changed
FNN based nominal NN	2	56
	2	68
RNN based nominal NN	3	56
	13	70

Table 2. The comparison of FNN and RNN for the nominal NN for 100 trajectories (failure rate)

Nominal NN (FNN)

- 3 layers fully connected
- 128 neurons
- ReLU
- RMSprop – 0.0008
- 4800 nominal parameters trajectories
- DGX-2 100 epochs

Robust NN (RNN)

- 2 layers fully connected + 2 layers Long Short-Term Memory (LSTM)
- 128 neurons
- ReLU
- RMSprop – 0.0008
- 2000 nominal + 7000 changed trajectories
- DGX-2 100 epochs

Simulations

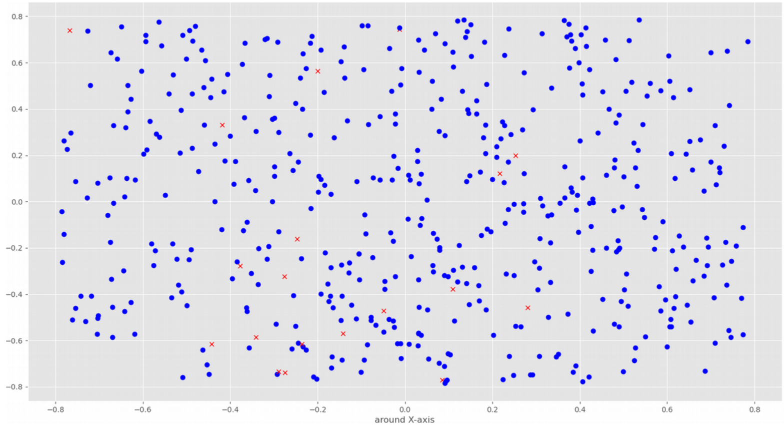
Different initial conditions for simulating the NNs
500 trajectories with six random states:

- Angular positions – $\theta 1$ and $\theta 2$
- Angular velocities – $\theta' 1$ and $\theta' 2$
- Wheel velocities – $\omega 1$ and $\omega 2$

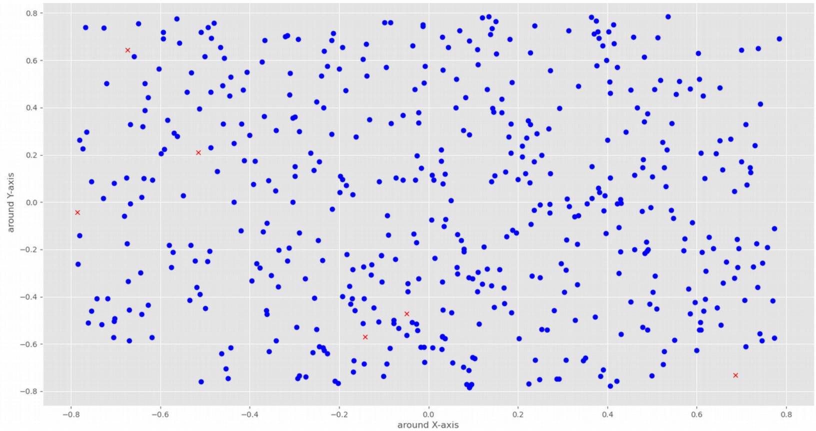
		Init_data 500 traj
Nominal NN	Nominal parameters	478 (95.6 %)
	Changed parameters	233 (46.6 %)
Robust NN	Nominal parameters	482 (96.4 %)
	Changed parameters	420 (84 %)

Table 3. The success rate Nominal NN and Robust NN for 500 random trajectories

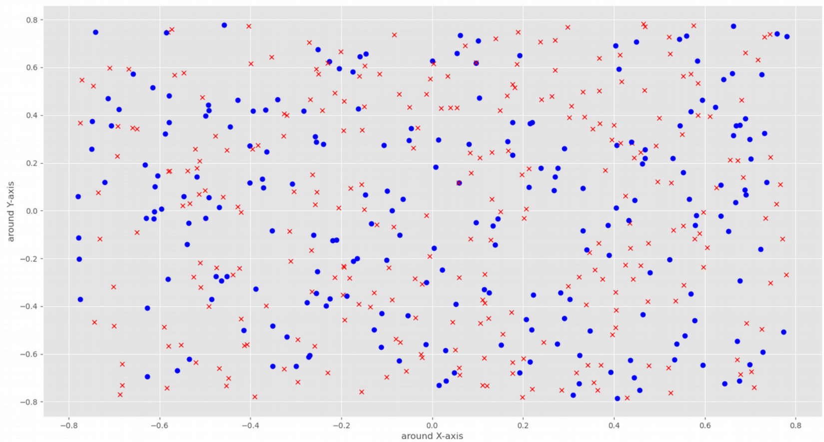
Simulations



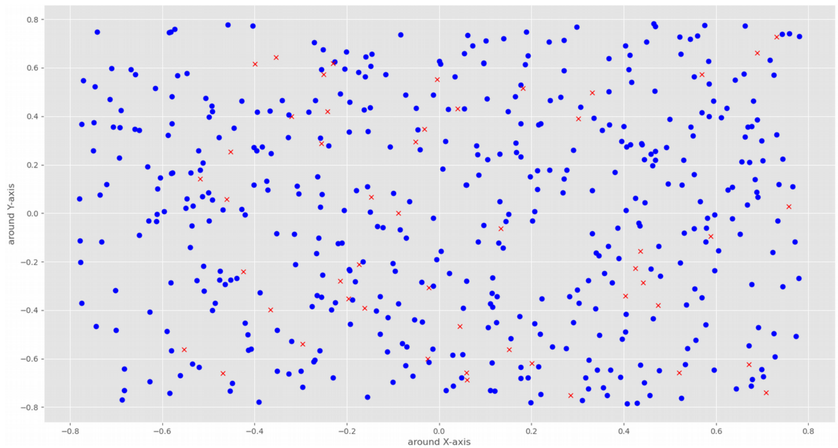
Nominal parameters, nominal NN



Nominal parameters, robust NN



Changed parameters, nominal NN



Changed parameters, robust NN

Figure 10. Simulation results (red dot – fail, blue dot – success)

Simulations

Nominal parameters:

- Nominal NN (478)
  - Mean – 1.16
  - Std – 0.16
- Robust NN (482)
  - Mean – 1.28
  - Std – 0.20
  -

Changed parameters:

- Nominal NN (233)
  - Mean – 1.18
  - Std – 0.20
- Robust NN (420)
  - Mean – 1.29
  - Std – 0.21

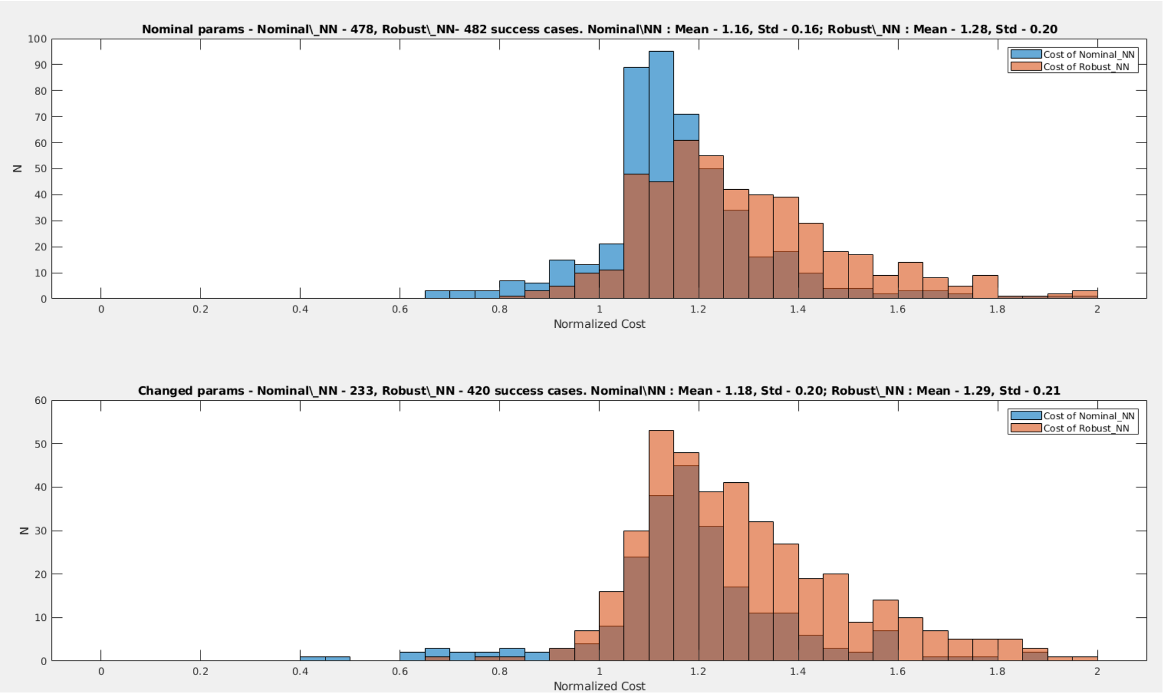


Figure 11. Histogram of costs of nominal NN and robust NN

Simulations

- Average of system parameters on x and y axes
- Mass – from 0 to 30 %
- Spring constants – from -20 to 0 %
- Pendulum friction coefficients – from -20 to 20 %

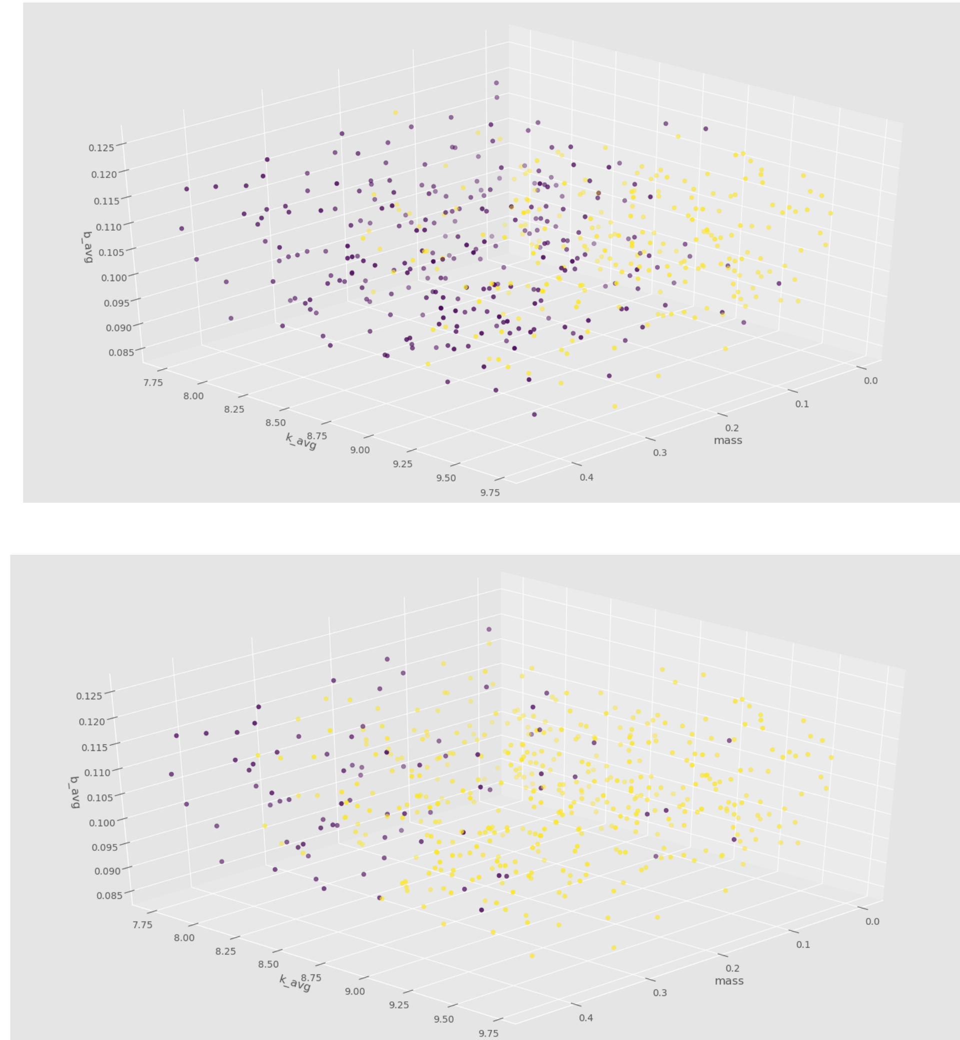


Figure 12. 3D plot of parameter simulation. (dark blue – fail, yellow – success)

Experiments

	Mass (grams)			
	0	100	200	300
Nominal NN	10	6	4	0
Robust NN	10	9	5	0

Table 4. The success rates of Nominal NN and Robust NN for 10 different trajectories

	Mass (grams)			
	0	100	200	300
Nominal NN (x1e-4)	21.65	26.27	28.64	29.46
Robust NN (x1e-4)	21.99	22.83	26.31	27.32

Table 5. Average cost of Nominal NN and Robust NN for 10 different trajectories.

Future work and Conclusion

- Conduct experiments with:
  - Putting balanced mass
  - Changed springs
  - Changed pendulum friction coefficient

- Reaction wheel integration improves VSA/VIA robots positioning accuracy
- Robust NN can deal with parameter uncertainties and has more accuracy comparing with nominal NN (~40 % more)

Thank you!

NAZARBAYEV
UNIVERSITY

References

Lasota, P. A., Fong, T., & Shah, J. A. (2017). A survey of methods for safe human-robot interaction. Now Publishers.

Baimyshev, A., Zhakatayev, A., & Varol, H. A. (2016). Augmenting variable stiffness actuation using reaction wheels. IEEE Access, 4, 4618-4628.

Muehlebach, M., & D'Andrea, R. (2016). Nonlinear analysis and control of a reaction-wheel-based 3-D inverted pendulum. IEEE Transactions on Control Systems Technology, 25(1), 235-246.

Belascuen, G., & Aguilar, N. (2018, June). Design, modeling and control of a reaction wheel balanced inverted pendulum. In 2018 IEEE Biennial Congress of Argentina (ARGENCON) (pp. 1-9). IEEE.

Türkmen, A., Korkut, M. Y., Erdem, M., Gönül, Ö., & Sezer, V. (2017, November). Design, implementation and control of dual axis self balancing inverted pendulum using reaction wheels. In 2017 10th International Conference on Electrical and Electronics Engineering (ELECO) (pp. 717-721). IEEE.

References

Baimukashev, D., Sandibay, N., Rakhim, B., Varol, H. A., & Rubagotti, M. (2020, July). Deep Learning-Based Approximate Optimal Control of a Reaction-Wheel-Actuated Spherical Inverted Pendulum. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1322-1328). IEEE.

Yutao Chen, Mattia Bruschetta, Enrico Picotti, and Alessandro Beghi. Matmpc- a matlab based toolbox for real-time nonlinear model predictive control. In 2019 18th European Control Conference (ECC), pages 3365–3370. IEEE, 2019.