2.152 Final Project Report: Adaptive Control for Microgravity Free-Flyers

Keenan Albee[1]

Abstract— **Manipulator-equipped robotic free-flyers have been proposed for spaced-based activities including assisting astronauts with daily tasks and serving as mobile workers in microgravity assembly scenarios. However, these robotic space systems must often operate in the face of uncertainty in various forms. One notable source of uncertainty for these systems is the parametric uncertainty that comes from grappling unknown or poorly characterized objects. For instance, after grappling a tool a robotic free-flyer must now move under a modified set of system dynamics where the mass, center of mass location, and moment of inertia of the end effector linkage have potentially shifted. For this particular uncertainty source, adaptive control emerges as a natural solution for following system reference trajectories since the parameters can be assumed to be constant.**

This final project will work toward producing an adaptive controller for a simple microgravity manipulator operating with a 3 degree of freedom (DoF) base and a single revolute linkage whose inertial properties are uncertain due to grappling an object. A feasible reference trajectory for nominal dynamics will be produced for the adaptive controller to follow. Additions are possible if the baseline project goal is achieved, including adding additional linkages or a full 6 DoF base, creating robust reference trajectories, or using a receding horizon trajectory generator in conjunction with the adaptive controller to e.g., account for changing operating conditions. Accounting for uncertainties in control design of these robotic spacecraft, including the manipulator-equipped Astrobee free-flyer currently aboard the International Space Station, will be essential for providing guarantees on system safety and improving performance.

I. INTRODUCTION

Interest in microgravity free-flyers—mobile satellite platforms that operate using both thrusters and internal reconfiguration like a manipulator arm—has grown considerably as new small satellites and on-orbit testbeds proliferate [8] [6]. These free-flyers, whose dynamics were extensively analyzed by Dubowsky and others in the late 1980s and 1990s for the shuttle program, are now of interest for a broader array of scenarios including active debris removal, on-orbit servicing, on-orbit assembly, and close-proximity astronaut assistance [1] [12]. The dynamics are more complex than the fixed-base case, including increased dimensionality, and underactuation and nonholonomicity (if thrusters are not used).

An interesting and useful scenario to consider is the control problem after these systems have grappled new, uncertain cargo. Naturally, these unknown, constant parameters invite the use of adaptive control as a solution to obtain better tracking despite not having initial knowledge of the system. This problem has parallels to the adaptive control of fixed robotic manipulators, but with additional dynamics complexity from the free-flying base link.

II. PROBLEM FORMULATION

The formulation here largely follows from Dubowsky et al. and from Virgili-Llop et al. [1] [11]. There exists a free-flying space manipulator, whose base is not rigidly fixed, resulting in a coupling between motion of the joints and motion of the base. In this case, a 2-link manipulator operating with a 3 degree of freedom (DoF) base will be considered.

Figure 1 is a representation of the free-flying manipulator system. The joints of this particular manipulator are all revolute and oriented out of the plane. The vector of joint angles is given by $\mathbf{q}$, with corresponding joint rates $\dot{\mathbf{q}}$. The base, with corresponding frame ${}^{0}\mathcal{F}$, has a position $\mathbf{r}_0$ and orientation $\mathbf{q}_0$, with corresponding velocities $\dot{\mathbf{r}}_0$ and $\boldsymbol{\omega}_0$, respectively. (Note that quantities without left superscripts are assumed to be in the inertial frarme, ${}^{I}\mathcal{F}$.)

The system position, velocity, and state vectors may be written (abusing notation a bit):

$$\mathbf{r} = [\mathbf{r}_0^\top \mathbf{q}_0^\top \mathbf{q}^\top]^\top \tag{1}$$

$$\mathbf{v} = [\dot{\mathbf{r}}_0^\top \boldsymbol{\omega}_0{}^\top \dot{\mathbf{q}}^\top]^\top \tag{2}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \end{bmatrix} \tag{3}$$

For the 3 DoF base planar 2-link system, these states are explicitly the following:

$$\mathbf{r} = \begin{bmatrix} r_1 & r_2 & q_0 & q_1 & q_2 \end{bmatrix}^\top \tag{4}$$

$$\mathbf{v} = \begin{bmatrix} \dot{r}_1 & \dot{r}_2 & \omega_0 & \dot{q}_1 & \dot{q}_2 \end{bmatrix}^\top \tag{5}$$

$$\tag{6}$$

The base and the links have masses $\{m_0, .., m_n\}$ and moments of inertia $\{I_{zz,0}, .., I_{zz,n}\}$, in addition to center of mass positions $\{\mathbf{r}_{CM,0}, .., \mathbf{r}_{CM,n}\}$. Geometrically, the links have lengths from joint-to-joint of lengths $\{l_1, .., l_n\}$, with an initial manipulator offset $\mathbf{l}_0$ from the base CM.

The dynamics, presented here in manipulator equation form in equation 7 are given by Dubowsky using a Lagrangian derivation [1]. However, Dubowsky uses the system center of mass (CM) position rather than the base position in his formulation—this makes the form of $\mathbf{H}$ sparser, but is a bit less intuitive when describing the robot state.

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}(\mathbf{q}, \omega_0, \dot{\mathbf{q}})\mathbf{v} = \boldsymbol{\tau} \tag{7}$$

[1] Ph.D. Candidate, Space Systems Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

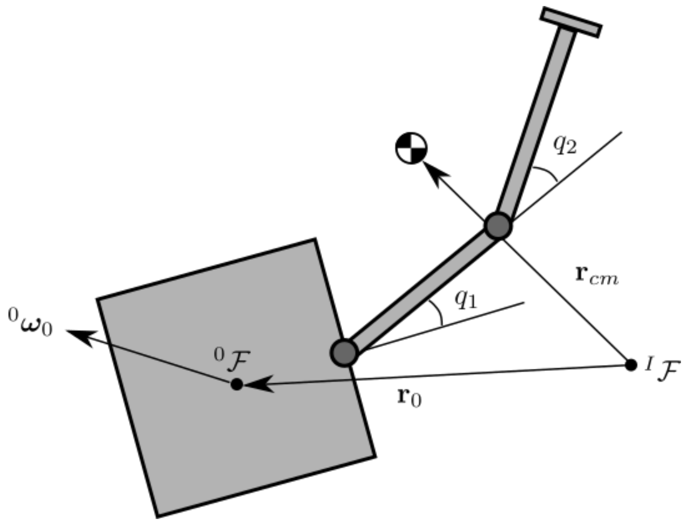

Fig. 1. The system sketch of a 2-link free-flying manipulator. The base is free to translate and rotate in the plane, resulting in a coupling between base and manipulator motion that is apparent in the manipulator equations.

Above, $\mathbf{H}$ is the inertia matrix, $\mathbf{C}$ is the matrix containing nonlinear terms (e.g., Coriolis), and $\boldsymbol{\tau}$ is the control vector. Here, the control vector consists of the following:

$$\boldsymbol{\tau} = \begin{bmatrix} {}^0\mathbf{F}_0^\top & {}^0\boldsymbol{\tau}_0^\top & \boldsymbol{\tau}_q^\top \end{bmatrix}^\top \tag{8}$$

$$\boldsymbol{\tau} = \begin{bmatrix} {}^0F_1 & {}^0F_2 & {}^0\tau_0 & \tau_1 & \tau_2 \end{bmatrix}^\top \tag{9}$$

Note that the thruster forces could also easily be written in ${}^0\mathcal{F}$ via a frame transformation. For this scenario, $\mathbf{H}$ has the following form:

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} M\mathbf{I}_{2\times 2} & \mathbf{M}_{r\omega} & \mathbf{M}_{rm} \\ \mathbf{M}_{r\omega}^\top & \mathbf{M}_{\omega} & \mathbf{M}_{\omega m} \\ \mathbf{M}_{rm}^\top & \mathbf{M}_{\omega m}^\top & \mathbf{M}_{m} \end{bmatrix} \tag{10}$$

(11)

M is the total system mass. $\mathbf{M}_\omega \in \mathbb{R}^{1\times 1}$ contains terms affecting $\boldsymbol{\omega}$, $\mathbf{M}_{\omega m} \in \mathbb{R}^{1\times n}$ is a coupling term that shows the effect of arm motion on base disturbance and $\mathbf{M}_m \in \mathbb{R}^{n\times n}$ captures the self-disturbance of the arm. The remaining terms are due to using the satellite base CM, rather than the system CM. Alternatively, the entire dynamics can be summarized in the robot manipulator format of equation 12, with terms grouped to explicitly represent impacts on the base, '0', and on the manipulator, 'm'. An additional $\mathbf{B}$ matrix is listed for mixing thruster outputs into force and torque commands, though it is not needed for this scenario.

A grappling scenario is considered, where some target has been grappled by the free-flyer and must be transported following some reference trajectory, $\mathbf{r}_{des}$. This results in uncertainty in equation 12, namely in the parameters involving the end effector. For simplicity and as a proof-of-concept, this formulation will consider a grappled *point mass*, meaning that there is a single unknown parameter: m_3, the mass of the grappled point mass payload perched exactly on the end of link 2. Figure 2 shows this scenario, where the payload in (b) can be thought of as a point mass.

III. APPROACH

In order of increasing complexity, some control approaches for this problem include simple PD position control, trajectory control, and adaptive control. This build-up is also discussed in Chapter 9 of Slotine and Li [7]. Ultimately, the adaptive control formulation is desirable as it is the only formulation that can account for the modified system dynamics.

A. Simple Position Control

PD position control aims to achieve a specified state vector using feedback on the system positions and velocities. Essentially, the system can be thought to have "virtual springs and dampers" that adjust its dynamics tune-ably based on the gain values and state reference. Proofs using virtual mechanical energy demonstrate that these systems are stable but may not have optimal tracking performance. For this system, a position controller is of the form may be used:

$$\boldsymbol{\tau} = -\mathbf{K}_p\mathbf{r}_{err} - \mathbf{K}_d\mathbf{v}_{err} \tag{13}$$

Stability is assured for positive definite $\mathbf{K}$ matrices. Note that this is a slight modification of from pure position control, in that a reference trajectory of $\mathbf{v}$ is included.

B. Trajectory Control

Building on position control, it is possible to achieve better tracking performance if a control law respecting a trajectory (containing one derivative beyond the state vector) is produced. For nonlinear systems, this is often accomplished via feedback linearization such that system accelerations follow ones prescribed. However, this assumes exact knowledge of the dynamics.

Using the manipulator dynamics of equation 12 and rearranging, a trajectory control strategy can be produced via:

$$\boldsymbol{\tau} = \mathbf{C}\mathbf{v} + \mathbf{H}[-\mathbf{K}_p\mathbf{r}_{err} - \mathbf{K}_d\mathbf{v}_{err} + \dot{\mathbf{v}}_{des}] \tag{14}$$

Note that a reference acceleration must be designated via this procedure, $\mathbf{v}_{\dot{des}}$. The gain matrices may be tweaked as before for differing system performance.

C. Adaptive Control

But what if there is uncertainty or a change in the system assumptions? One approach is to compute reference trajectories fast enough such that these reference trajectories are themselves a method of control, called model predictive control (MPC). However, real-time computation is sometimes infeasible and nonlinear variants offer scant guarantees on robustness or convergence [5]. Tube-based robust MPC is being explored further for this purpose [3] [4]. However, traditional reference-tracking control with guarantees on performance may be desirable.

Adaptive control, which uses sliding mode control to adapt parameters online and converge to a sliding surface

$$\begin{bmatrix} \mathbf{M}_0 & \mathbf{M}_{0m} \\ \mathbf{M}_{0m}^\top & \mathbf{M}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_0 \\ \ddot{\mathbf{q}}_m \end{bmatrix} + \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{0m} \\ \mathbf{C}_{0m}^\top & \mathbf{C}_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\mathbf{q}}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{bmatrix} + \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_{0m} \\ \mathbf{B}_{0m}^\top & \mathbf{B}_m \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_m \end{bmatrix} \tag{12}$$

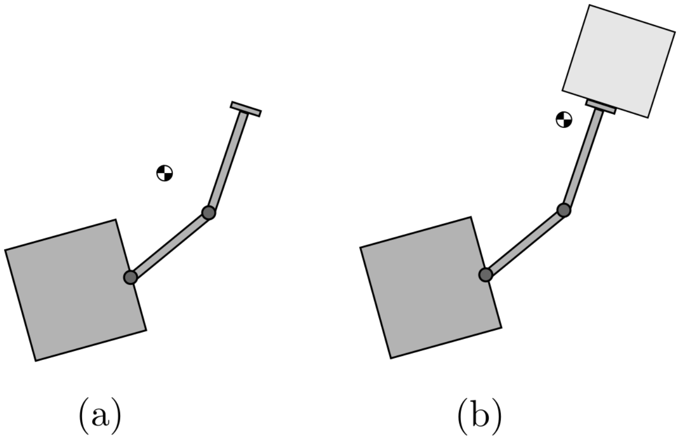

Fig. 2. In (a), the nominal system has a finely-tuned model and controller, accounting for the deterministic known parameters and free-flying dynamics. In (b), the system has grappled a payload of unknown mass, adding an unknown parameter to the model, m_3. The nominal system model has surely changed.

is appropriate. Previous investigations using adaptive control for space applications have been demonstrated for both rigid and flexible link manipulators, though notably with only *fixed bases* in [9] [10]. However, work by Senda et al. first adapted some of the developments by Slotine and Li to free-flying adaptive control. In the late 1990s, Guelman et al. simplified the task of parameter estimation to require only estimation of the *relative influence* of grappled payload on the dynamics. Notably, other interesting approaches consider e.g., multiple arms or other control methods [2].

However, these approaches rarely provide a full view of their dynamics and controller formulations which is unfortunate particularly for Senda et al. Senda's approach uses an adaptive control approach based directly on Slotine's fixed-base adaptive controller, which relies on a linear parameterization of the dynamics in terms of the adaptation parameters [7]. However, the free-flying dynamics are much more complicated and higher-dimensional than the fixed-base dynamics, so determining reasonable parameterizations is extremely challenging (if even feasible). Senda does not discuss these details and presents some limited 2D simulations.

An adaptive controller similar to Senda's and the generic manipulator equation formulation of Slotine and Li is presented here, with an attempt at linear parameterization. The uncertain parameter vector, $\mathbf{a}$ is simply

$$a = [m_3] \tag{15}$$

An adaptation and control law must be produced so that the unknown parameters can be tweaked during operation. Assuming linearly parameterized dynamics, a sliding control approach can be taken. For a 2nd order system, a sliding variable $\mathbf{s}$ is introduced:

$$\mathbf{s} = \mathbf{v}_{err} + 2\lambda_0 \mathbf{r}_{err} \tag{16}$$

where λ is used to define convergence on the sliding surface. Reference values of $\dot{\mathbf{v}}_r$ and $\mathbf{v}_r$ are produced according to the desired trajectory and current tracking error:

$$\dot{\mathbf{v}}_r = \dot{\mathbf{v}}_{des} - 2\lambda_0 \dot{\mathbf{e}} \tag{17}$$

$$\mathbf{v}_r = \mathbf{v}_{des} - 2\lambda_0 \dot{\mathbf{e}} \tag{18}$$

The next step is choosing a set of parameters (perhaps more complex than the actual underlying uncertain parameter(s)) that can be linearly combined with nonlinear functions to produce $\dot{\mathbf{v}}$. For this 10-dimensional nonlinear system, this is a real challenge. The approach taken here uses the MATLAB Symbolic Toolbox to group nonlinear terms and separate out just m_3—however, this still leaves m_3 within the nonlinear terms. This work simply leaves m_3 within the parameter vector and uses the latest estimate $\hat{a}$, and using the resulting matrix for adaptation. This likely forfeits convergence guarantees of the approach, though a more in-depth analysis is required. It would also be worth looking into other choices of adaptation parameters that might allow for a linear parameterization of the plant dynamics. The quasi-parameterization can be written:

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{v}}_r + \mathbf{C}(\mathbf{q}, \omega_0, \dot{\mathbf{q}})\mathbf{v}_r = \mathbf{Y}(\mathbf{r}, \mathbf{v}, \mathbf{v}_r, \dot{\mathbf{v}}_r, \hat{a})a \tag{19}$$

A control law to feedforward based on the best estimate, along with feedback on the converging sliding variable, is given as equation 21 with accompanying adaptation law equation 22:

$$\boldsymbol{\tau} = \mathbf{Y}\hat{a} - \mathbf{K}_D \mathbf{s} \tag{20}$$

$$\dot{\hat{a}} = -\Gamma \mathbf{Y}^\top \mathbf{s} \tag{21}$$

$$\tag{22}$$

where Γ influences adaptation rate and $\mathbf{K}_D$ is used to show Lyapunov convergence. However, a more detailed convergence analysis is necessary since $\mathbf{Y}$ is a function of $\hat{a}$ here.

IV. RESULTS

The free-flying manipulator dynamics were implemented in MATLAB, using a wrapped version of the Spacecraft Robotics Toolkit (SPART) for the forward dynamics. [11] MATLAB's ODE23 numerical integration tool was used for system propagation: the reference trajectory and adaptive tracking controller were integrated into this framework. The analysis here is for unconstrained motion, though one could conceivably create more complicated reference trajectories or use a real-time control approach that considers e.g., obstacle constraints.

Reference trajectories were created for a simple step-response, and a more complex base movement with sinusoidal arm movements. Only results from the complex movement are shown here. The full trajectory was obtained using simple finite differencing of the prescribed position commands. Both of these trajectories were tested with tuned versions of the aforementioned controllers. Finally, the complex trajectory was tested using the controllers with modified inertial properties due to grappling an object: these modified properties were not known beforehand, so gains (and parameter estimates) are unchanged, with the notable exception of the adaptive controller. The reference trajectories are made “sufficiently fast” so that model error and nonlinearity becomes significant in system performance.

For both cases, $m_0 = 10$ kg, $m_1 = m_2 = 1$ kg, $I_{zz,0} = 0.1$ kg^2-m^4, $I_{zz,1} = I_{zz,2} = 0.05$ kg^2-m^4 with $l_1 = l_2 = 0.3$ m, with CM at the link centroids. However, for the modified case a new mass is added to the end of link 2, with $m_3 = 8$ kg, a significant mass fraction compared to the base.

A. Simple Position Control

The PD controller was tuned for the nominal case, but does not have perfect tracking—it is only a position/velocity controller and cannot account for nonlinearities.

B. Trajectory Control

The trajectory controller was also tuned for nominal tracking performance.

C. Adaptive Control

The adaptive controller was the only controller used that had the ability to actively compensation for the change in mass parameters.

1) Modified End Effector Parameters:

V. CONCLUSIONS

APPENDIX

ACKNOWLEDGMENT

REFERENCES

[1] Steven Dubowsky and Evangelos Papadopoulos. The Kinematics, Dynamics, and Control of Free-Flying and Free-Floating Space Robotic Systems. *IEEE Transactions on Robotics and Automation*, 1993.

[2] Ying Hong Jia, Quan Hu, and Shi Jie Xu. Dynamics and adaptive control of a dual-arm space robot with closed-loop constraints and uncertain inertial parameters. *Acta Mechanica Sinica/Lixue Xuebao*, 30(1):112–124, 2014.

[3] Brett T Lopez. Adaptive Robust Model Predictive Control for Nonlinear Systems. (2019), 2019.

[4] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano. Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance. *Aerospace Science and Technology*, 77:585–594, 2018.

[5] Tomasz Rybus, Karol Seweryn, and Jurek Z. Sasiadek. Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (NMPC). *Journal of Intelligent & Robotic Systems*, 85(3-4):491–509, 2017.

[6] Alvar Saenz-Otero and David W Miller. *Design Principles for the Development of Space Technology Maturation Laboratories Aboard the International Space Station*. PhD thesis, 2005.

[7] Jean-Jacques E Slotine and Weiping Li. On the Adaptive Control of Robot Manipulators.

[8] Trey Smith, Jonathan Barlow, Maria Bualat, Terrence Fong, Christopher Provencher, Hugo Sanchez, and Ernest Smith. Astrobee: A New Platform for Free-Flying Robotics on the ISS. 2016.

[9] Steve Ulrich and Jurek Z. Sasiadek. Modified Simple Adaptive Control for a two-link space robot. *Proceedings of the 2010 American Control Conference, ACC 2010*, pages 3654–3659, 2010.

[10] Steve Ulrich, Jurek Z. Sasiadek, and Itzhak Barkana. Modeling and direct adaptive control of a flexible-joint manipulator. *Journal of Guidance, Control, and Dynamics*, 35(1):25–39, 2012.

[11] Josep Virgili-Llop, Jerry V Drew Ii, and Marcello Romano. SPACECRAFT ROBOTICS TOOLKIT: AN OPEN-SOURCE SIMULATOR FOR SPACECRAFT ROBOTIC ARM DYNAMIC MODELING AND CONTROL.

[12] Kazuya Yoshida, Kenichi Hashizume, and Glnd Satoko. Zero Reaction Maneuver : Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm H b Hbm. pages 441–446, 2001.

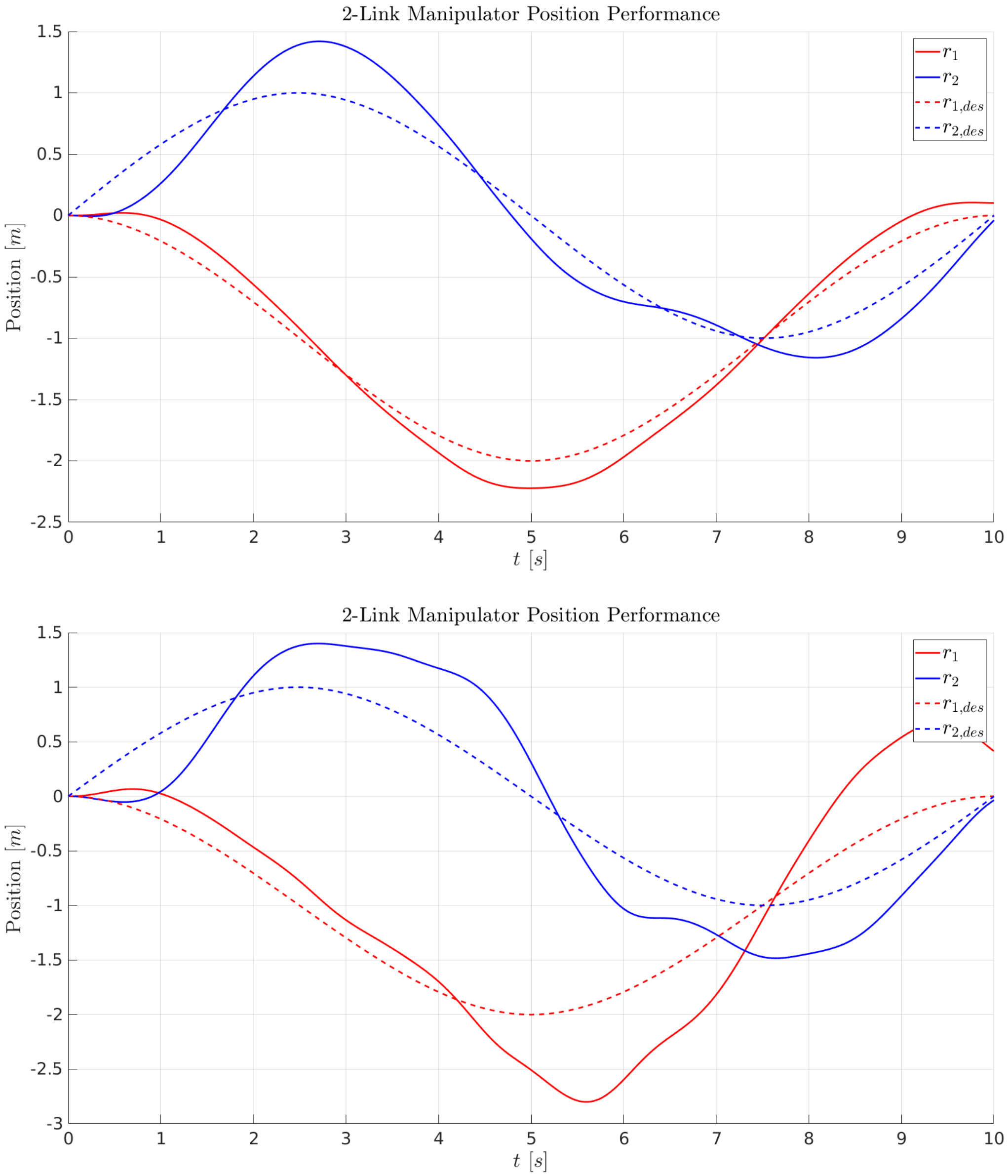

Fig. 3. This is a tiger.

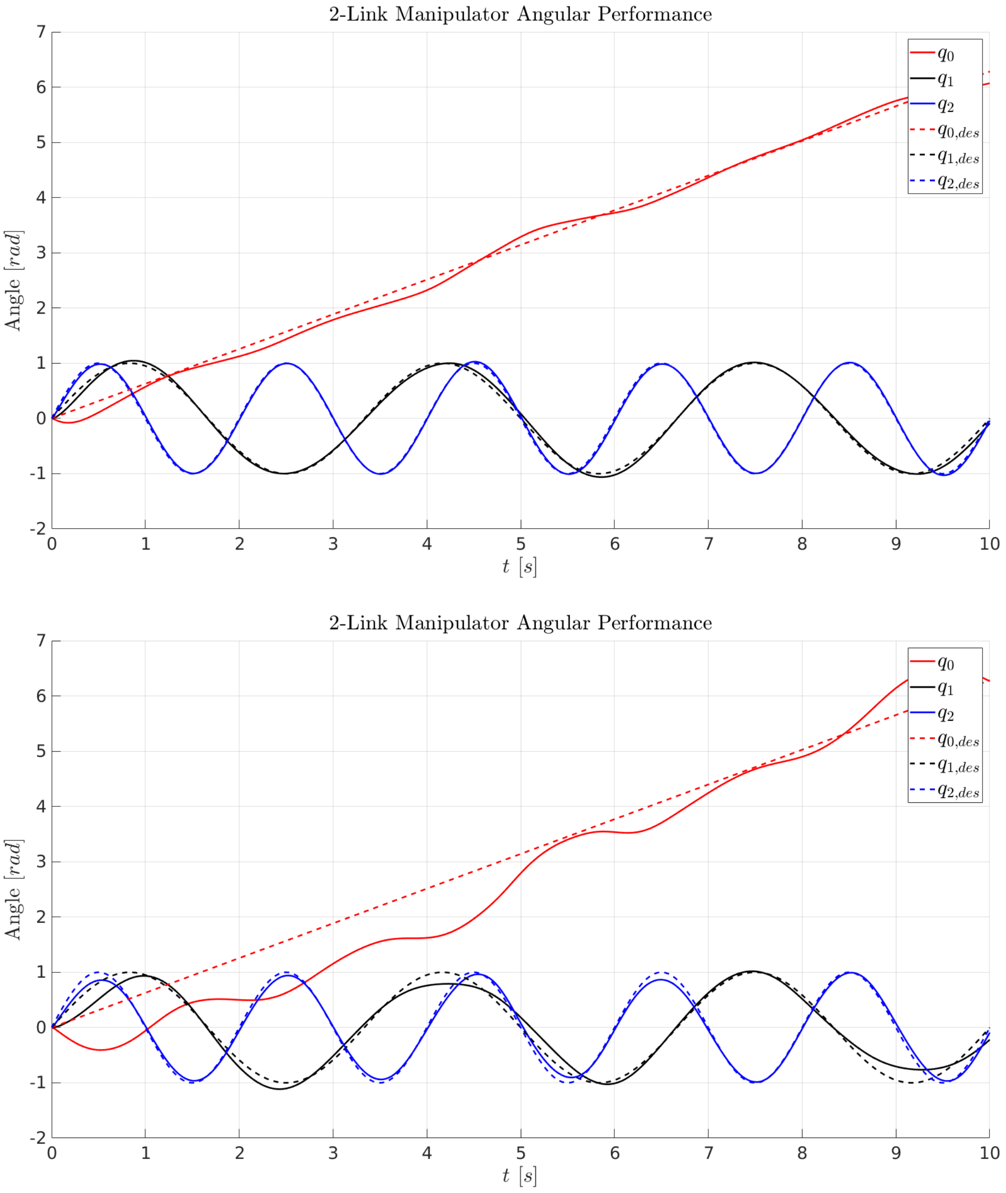

Fig. 4. This is a tiger.

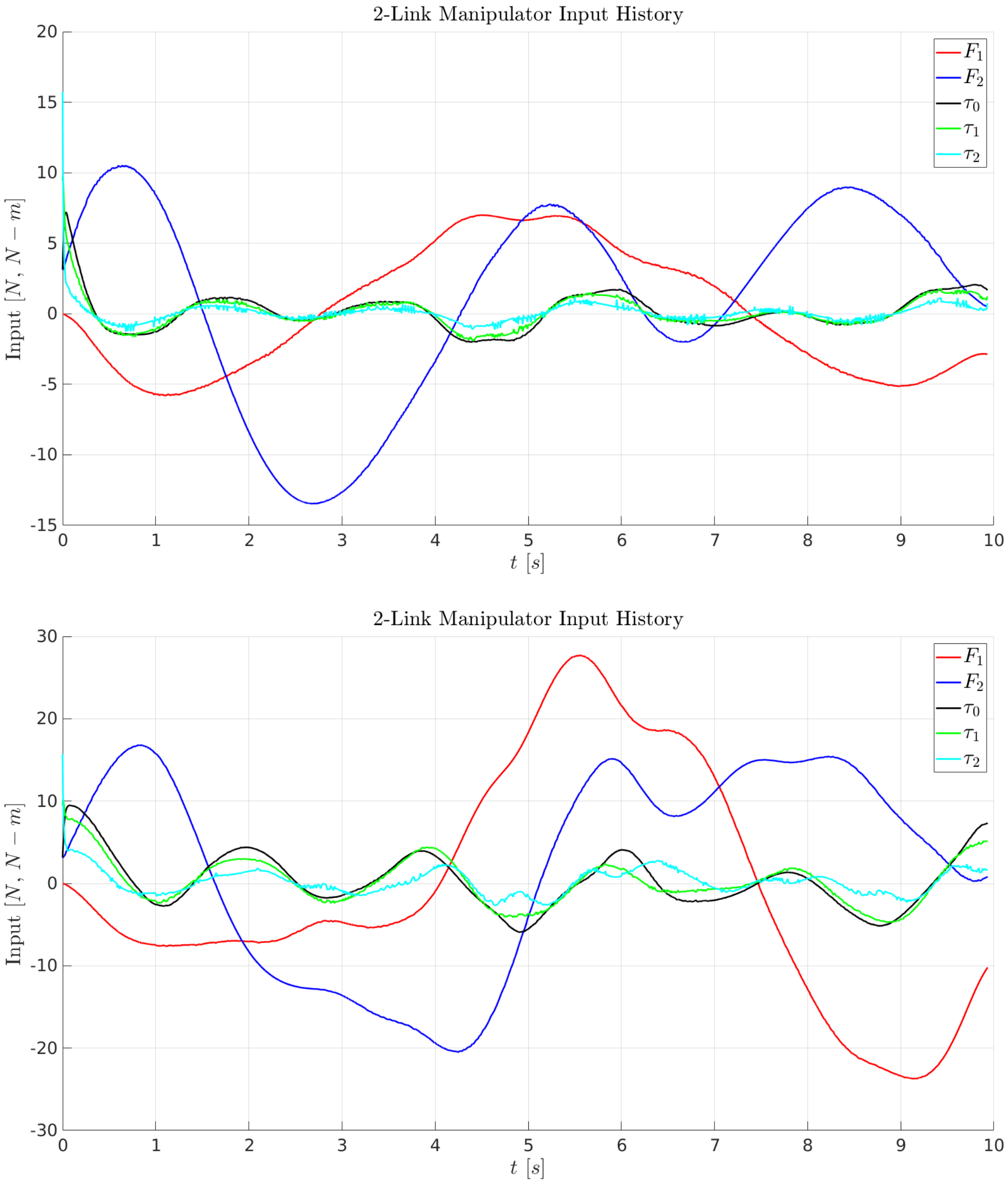

Fig. 5. This is a tiger.

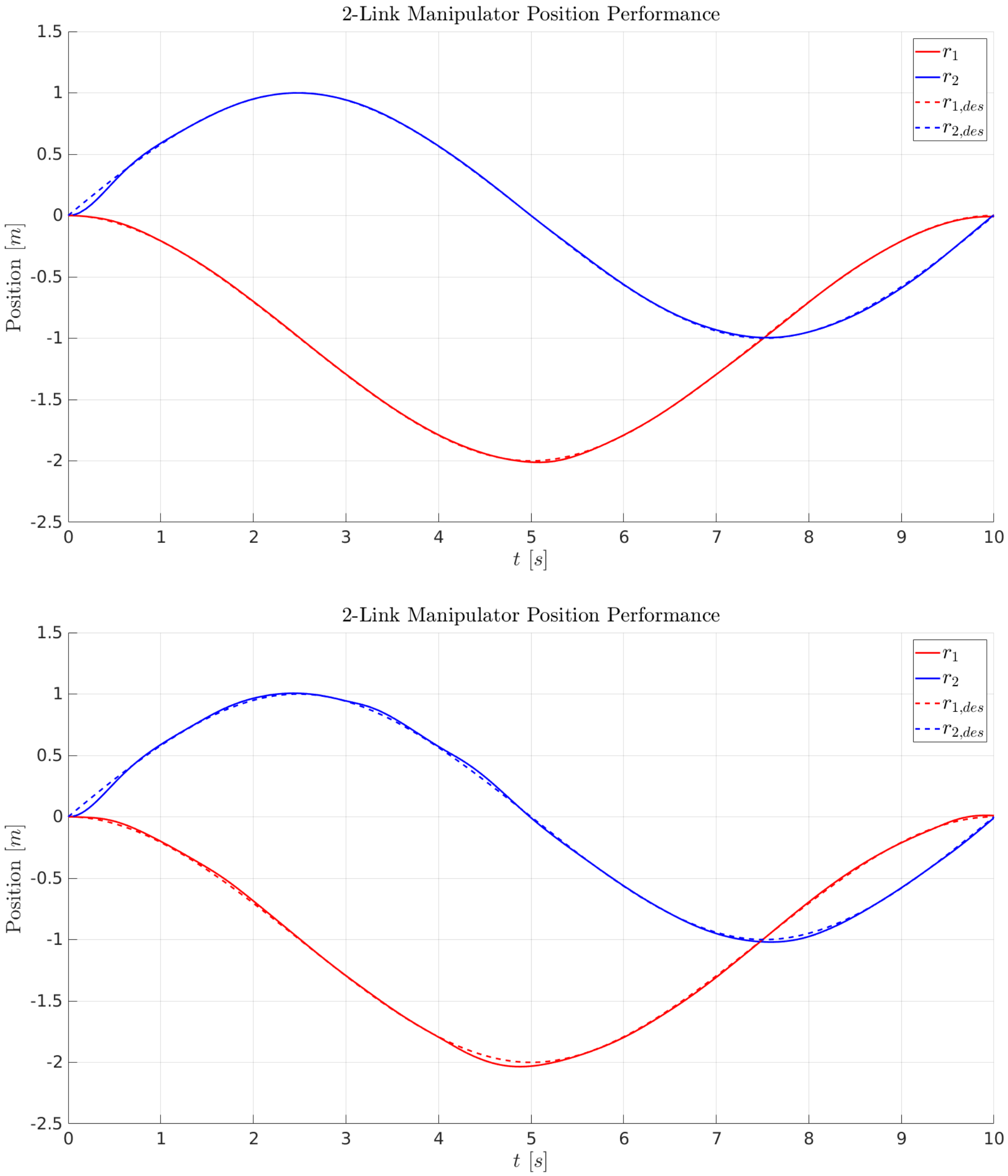

Fig. 6. This is a tiger.

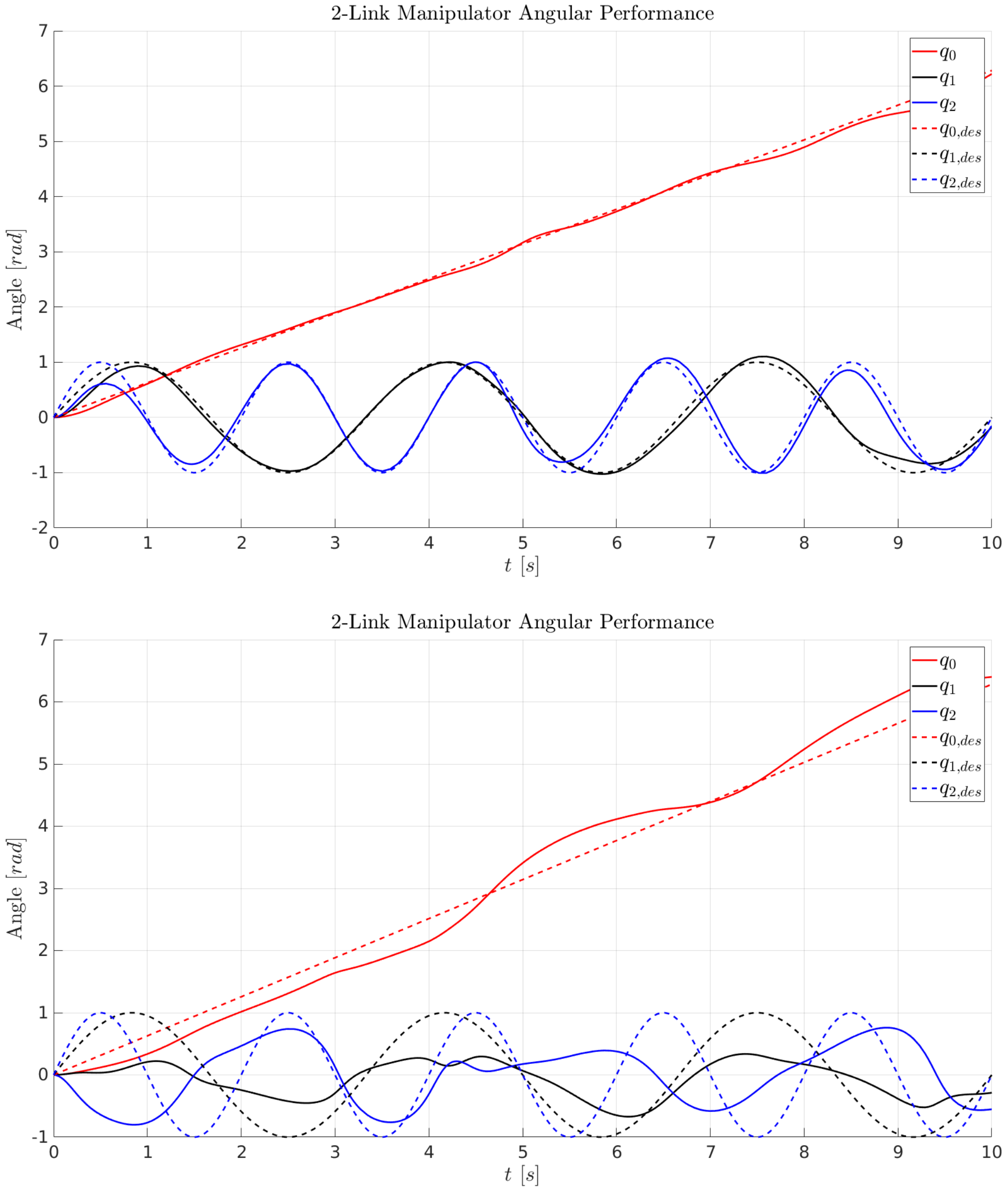

Fig. 7. This is a tiger.

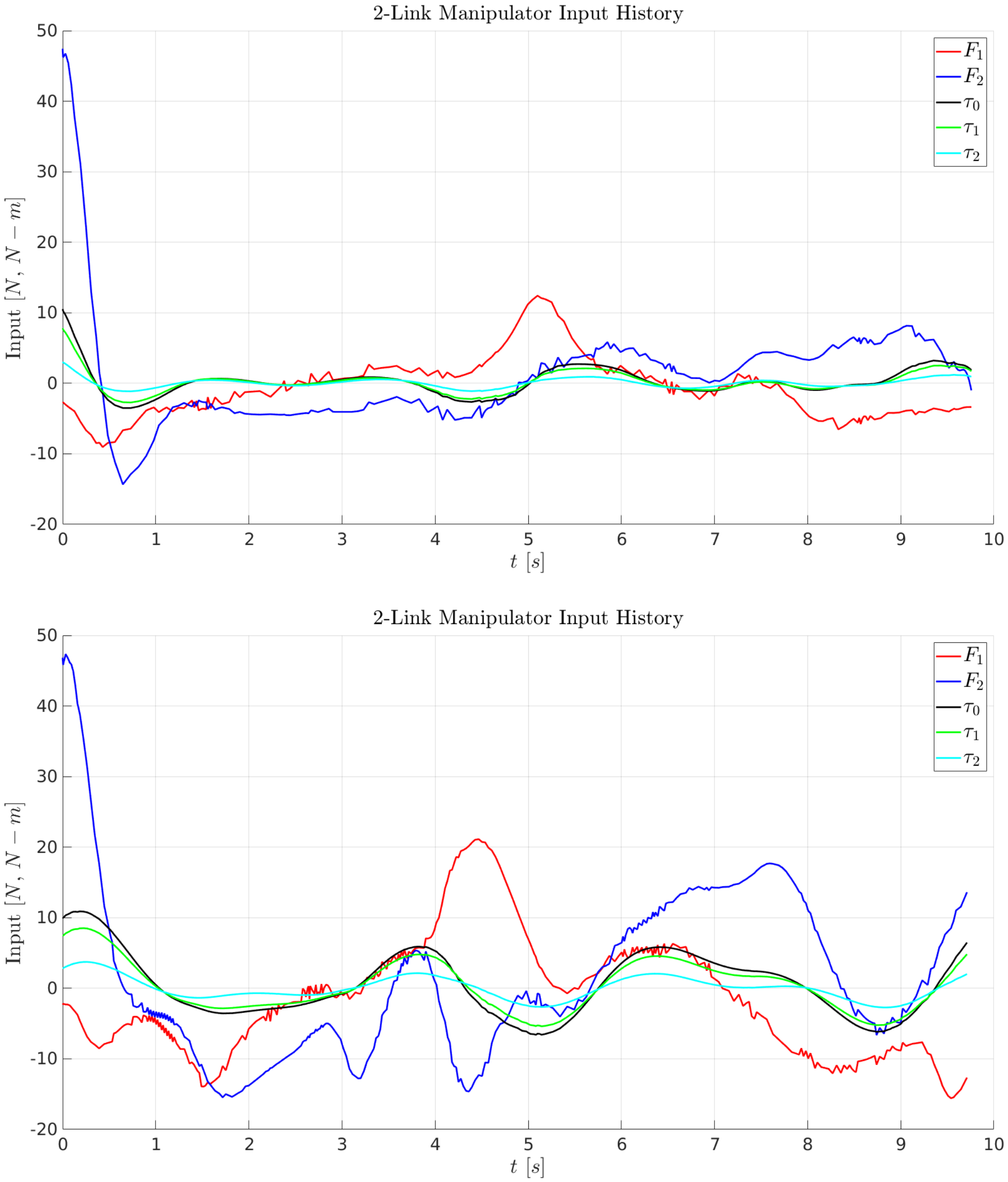

Fig. 8. This is a tiger.

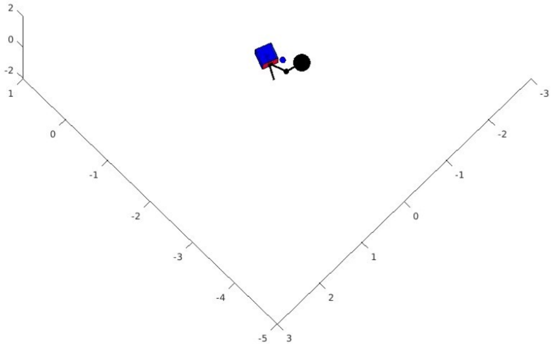

Fig. 9. This is a tiger.

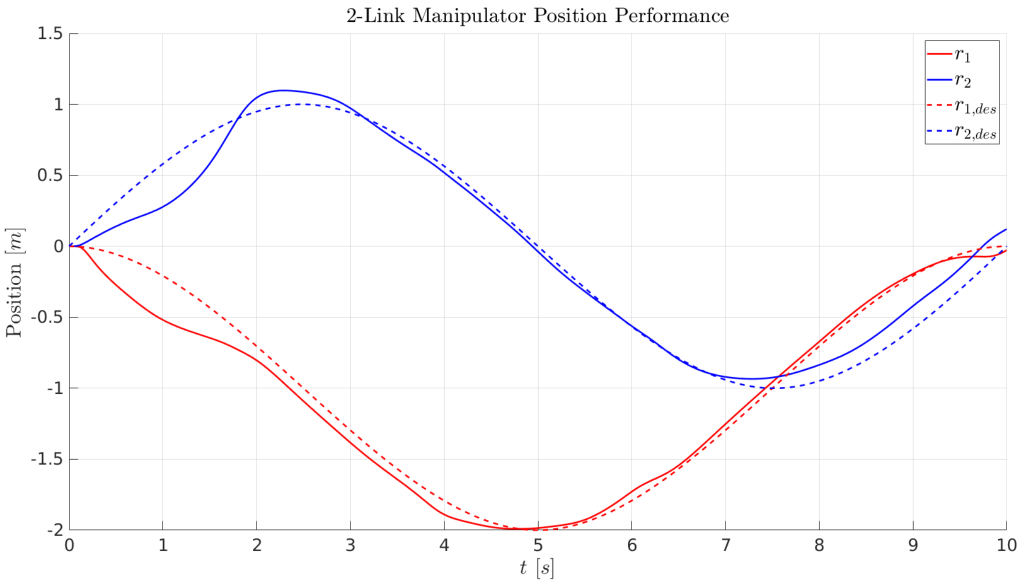

Fig. 10. This is a tiger.

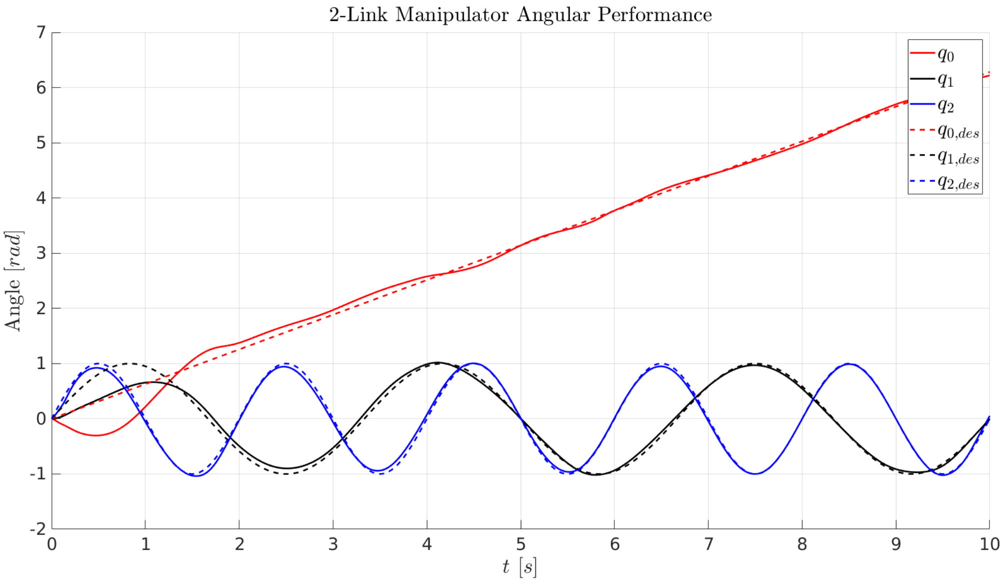

Fig. 11. This is a tiger.

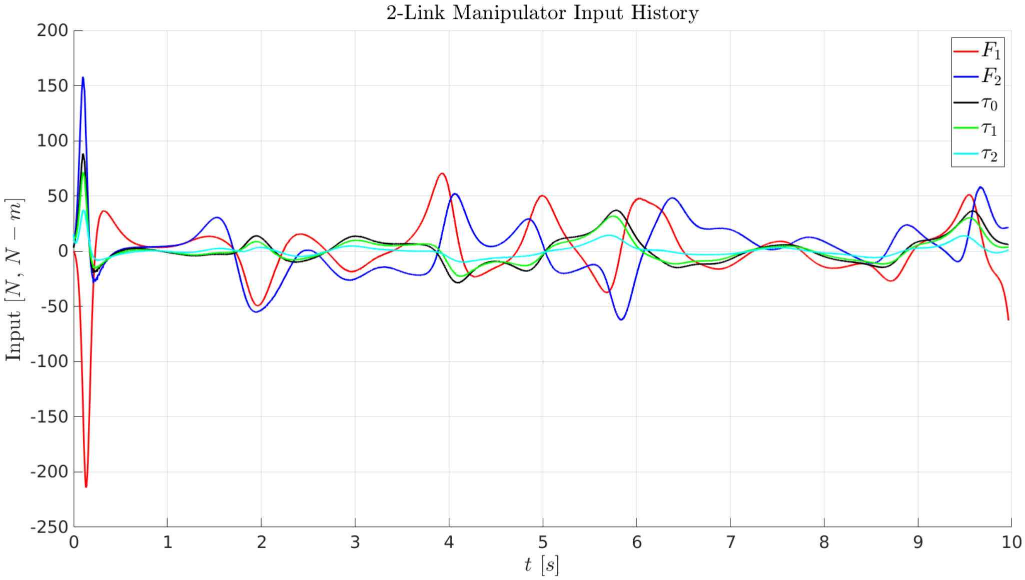

Fig. 12. This is a tiger.

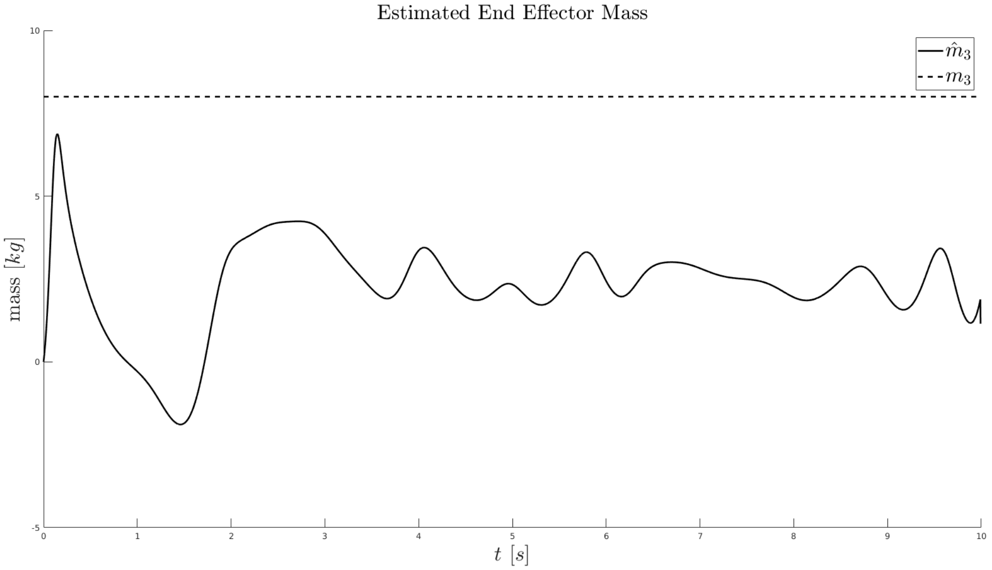

Fig. 13. This is a tiger.