Provide your results, discussion, and Matlab code for the following questions in a single PDF.

1. In this problem, you will implement the time-varying constraint tightening approach to robust MPC developed in [1]. Specifically, you will implement the numerical example presented in Section 2.5.1 of [1] to recreate Fig. 2-2. It is expected that you read and understand the material presented in Section 2 of [1] in order to understand the details of this specific approach to robust MPC. While all of the details needed to implement the numerical example are provided in [1], use the following steps to guide your implementation.

   (a) First, define the A, B, C, and D system matrices, the prediction horizon N, and the initial condition $x_0 = 0$ based on the information provided in [1].

   (b) Next, design the nilpotent candidate feedback controller using the `acker` command in Matlab. Note that the `acker` command assumes a negative feedback controller while our work assumes a positive feedback controller. Check your candidate controller with the one provided in [1].

   (c) Now define your sets. Define the original output constraint set $\mathcal{Y}$ as a polyhedron using the MPT. Similarly, define the bounded disturbance set $\mathcal{W}$. Use a for loop to iteratively compute the tightened output constraint sets $\mathcal{Y}(k)$ for all $k = 0, ..., N$ based on equations (2.8) and (2.9) in [1].

   (d) You are now ready to define your controller using Yalmip in Matlab. When formulating this controller make sure that you use the same cost function and terminal constraint as specified in [1]. Additionally, add an if statement in your controller formulation so that you can define a variable (for example `robustFlag`) where if `robustFlag` $= 0$ then you use the original output constraints $y(k) \in \mathcal{Y}$ and if `robustFlag` $= 1$ then you use the tightened output constraints $y(k) \in \mathcal{Y}(k)$.

   (e) Finally, implement your controller in closed-loop simulation. Use a total of 100 simulations where each simulation is 100 time steps long. Use the `rng` command to set the random number generator seed for your simulations and the `rand` to help create your random disturbance. Use these simulations for the nominal and robust controller to recreate Fig. 2-2 from [1] as close as possible, including the 'o' to denote when the optimization problem becomes infeasible for the nominal MPC case without constraint tightening. Note that we are only plotting the first state and your results might be slightly different due to the difference in random disturbances (but the overall trend/behavior should be the same). In addition to the state trajectories, also provide plots of the input trajectories.

   (f) For each of the 100 simulations, also compute the total operating cost as the sum of the stage cost at each of the 100 time steps based on the stage cost used in your controller formulation. Provide the mean and maximum total operating cost across the 100 simulations. You will compare these values to those from the tube-based MPC approach in the following problem.

2. For this problem, you will implement the tube-based approach to robust MPC developed in [2]. You will use the exact same system, candidate control law, and control objective as Problem 1 so that the only difference is how robustness is achieved through constraint tightening. Once again, you are expected to have read and understood the details of this specific approach to robust MPC provided in [2]. Use the following steps to guide your implementation.

(a) Define the system matrices, prediction horizon, initial condition, and candidate feedback controller as in Problem 1. Instead of defining an output constraint set, define separate state and input constraints sets, $\mathcal{X}$ and $\mathcal{U}$. Define the disturbance set $\mathcal{W}$.

(b) Next, compute the RPI set using the minRPI set code provided on eLearning. Plot this RPI set and play with different values of `epsilon` to ensure that you have an RPI set that is not heavily dependent on the specific choice of `epsilon`.

(c) Compute the tightened state and input constraint sets and provide the corresponding bounds on the nominal state and inputs. Comment on the relative size of the original and tightened sets.

(d) Formulate your controller based on the tightened state and input constraints and the unique initial condition constraint discussed in [2].

(e) Implement this controller in closed-loop simulation with only the robust formulation (you don't need to do the nominal and robust cases like in Problem 1). Remember that you actually need to implement the candidate control law based on the difference between the measured and nominal state (unlike in Problem 1 where the input determined by MPC is applied directly to the system). Provide both the state and input trajectory plots.

(f) Finally, compute the total operating cost for each of the 100 simulations and provide the mean and maximum. Compare these to the results from Problem 1. Is there a significant difference and is this difference qualitatively supported by comparing what you see in the state and input trajectories? Use this to compare the pros and cons of the tube-based and time-varying constraint tightening approaches.

References Cited

[1] Arthur George Richards. *Robust Constrained Model Predictive Control*. PhD thesis, Dept. Aeronautics and Astronautics, MIT, 2005.

[2] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances. *Automatica*, 41:219–224, 2005.