Study and Understand Hw robust

Taozheng Yang

Contents

Lessons learned from the file (Verify MUSSV)

In order to use the high-level command mussv, the input argument needs to follow the specific rule stated in the official documentation to create the block structure.

The output of the mussv, the muinfo gives us a comprehensive result about the upper bound information (the newlin/young method consisting finding the scailing beta, D, and G, the combination of which is smaller than 1 would certify the upperbound; the semidefinite verification of the upperb ouds is connected to the lecture slides, which seays the G term is included to shift the complex disk to cover the real term in the delta) and lower bound information (VDelta, which is for the lower bound, the product of VDelta and M would make the det(I-M*VDelat = 0), and the system will be unstable, thus we could find the structured delta for the system to be unstable).

If the block is purely complex, then the general formulations of the upperbounds and lowerbounds can be simplified, which is consistent with the lecture notes. Additionally, for three blocks, where one contains another, the largest block would give the smallest minimum singular value, and the reciprocal of which is the uppper bound, and thus the biggest upperbound among the three.

Lessons learned (DestabilizingPerturbation)

loopsens provides a complete analysis result of a system's sensitivity properties comprised by a plant and a negative feedback controller. The output of the loopsens could be extracted to do other analysis and proof of some theories such as the small gain theory.

The smallest plant input uncertainty causing the system to be unstable can be calculated by using the singular value decomposition, where we can obtain the delta, and det(I-M*delta) = 0, and the system is on the fringe of being unstable. Notice the M = TiNorm

The instability of the system can also be visualized by probing the poles of the uncertainty perturbed system (where the perturbation can be a dynamic system as well as a complex matrix mimicing the dynamic system).

Lessons learned (UncertainModelling)

This tutorial teaches me how to create uncertain objects by specifying their nominal value and range or percentage of variability. Additionally, how to sample certain number of points from the uncertain objects created.

Additionally, how to use the ultidyn function to generate uncertain linear time invariant dynamical system, and use the usample to extract certain number of sample variables from the uncertain dynamical system. Last but not the least, how to use the lftdata to decompose an unceratin system into its consitituents, the nominal part and the uncertain parts. And how to use the lft to form back the original uncertain system with thenominal part and the uncertain part.

Lessons learned (lftExplore)

From the tutorial, I know how to model a system with uncertainty using lftdata, and what are their matrix representations look like. Autosimplify can be used to simplify the LFTs.

Lessons learned (WorstCaseAnalysisIntroduction)

This tutorial presents the basic and fundamental usage of the built-in analysis tool such as robuststab and robstab, both are used to analyze the robustness of the system, and give a detailed information about the unstable pole frequency, the percentage of each uncertainty object's contribution, and the system's tolerance of the uncertainty..

Additionally, we can use the built-in function wcgain to obtain the worst case gain and the corresponding offending uncertainty leading to the worst case, with which we can construct the nominal and worst case systems to compare their performances such as frequency response or step response.

Lessons learned (stateSpaceParameterUncertainty)

This tutorial presents an instance of a mass-spring-damping system with uncertain parameters, and thus uncertain state space. Basically, I learnerd to know how to construct an uncertain state space model on top of the nominal values of all the state space matrices, and finally visualize the system's performances.

Lessons learned (dcmotor_demo1)

This script creates a seires of unceratin parameters affecting the performance of the dc motor, and construct the transfer function from the input to the output. After that, the script only analyzes the angular speed transformation by aoolying lft to the uncertain system. Finally, the bode plot and step response of the open loop nominal angular speed and uncertain angular speed are generated.

Lessons learned (rsrpmu)

This script presents various ways to analyze an uncertain system's robustness, through both high level command and low level command. After we modeled the plant and the controller, we can use the command robuststab to analyze the robustness of the negative feedback system.

It also shows how to use connect and makeweight to combine multiple competing objectives into one objective, and similarly, the robuststab can be utilized to generate the robustness report. We could further extract information about the destabilizing uncertain parameters closest to their nominal values, and find the destabilizing frequency as well.

The mu analysis can generate the same result as that of the high-level command robuststab. Basically, using lftdata to extract the delta, which is only related to the uncertainty channels. Subsequntly, we use the mussv to extract the mu bound, which can be used to calculate the upper bound and lower bound.

Lessons learned (FlightControlExWithSSV)

This script presents an instance of a flight control. Basically, an uncertain plane model is created, and it is further processed to form a plant, an uncertain state space. Then, the controller is presented, and a feedback system is constructed. Finally, both methods mentioned in rsrpmu are implemented to find the robustness of the system.

Lessons learned (mimoMotivate mimoMotivateResolveMU)

These two scripts show that even though a system has good nominal performance and good robust stability does not gurantee robust performance for a MIMO system, but not for a SISO system. mimoMotivateResolveMU uses mu analysis to show this phenomenon.

Published with MATLAB® R2017b