Nonlinear Control And Analysis
Home Taken Exam 1b

HAN University of Applied Sciences

Master of Engineering Systems

Embedded Control Module

Date	Change Summary	Authors
2024.04.05	Fixed the typo in Step 8: “to find … matrix P”	Amin Mannani
2024.03.17	Completed the document	Amin Mannani
2024.03.14	Created the document	Amin Mannani

Table of Contents

Assignment

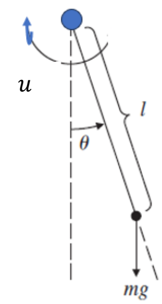
Learning Goals

1. Expressing nonlinear models (of robots) in linear parameterization form
2. Designing adaptive control for the nonlinear (robotic) systems
3. Analyzing the convergence of the system error and parameter identification using different algorithms.
4. Providing an implementable work

Problem Statement

Consider the model of a friction-free (or undamped) pendulum which is driven by the torque $u(t)$ at its hinge. Denote the mass with m, the inertial with I, the length of the rod with l and the angle of the rotation with $\theta(t)$. This can be also considered as a single-link manipulator.

$$\boxed{\begin{gathered} I\,\ddot{\theta}(t) + mgl\sin\theta(t) = u(t) \\ I = 10\ [kgm^2] \\ mgl = 10\ [Nm] \end{gathered}}$$



In this assignment we want to design an adaptive controller for this system. We assume that we do not know the values of I and mgl above, and the adaptive controller estimates these values, too.

Design Steps

To design an adaptive controller go through the following steps, which are mentioned in the lecture notes for any linearly parametrizable robotic system, and apply them to the case of the above model.

1. Rewrite the model in the generic form $M\,\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{u}(t)$ to distinguish $M, \boldsymbol{h}$ and $\boldsymbol{q}(t)$**,** which have been used in the general formulation.
2. Choose the inverse dynamics control $\boldsymbol{u}(t) = \widehat{M}\,(\boldsymbol{q})\big(\ddot{\boldsymbol{q}}^d - K_0\boldsymbol{e} - K_1\,\dot{\boldsymbol{e}}\big) + \,\widehat{\boldsymbol{h}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, where $\widehat{M}$ and $\widehat{\boldsymbol{h}}$ are the estimates of M and $\boldsymbol{h}$, respectively, and $\boldsymbol{e}(t) = \boldsymbol{q}(t) - \boldsymbol{q}^d(t)$ is the tracking error.
3. Derive the resulting dynamics of the error $\boldsymbol{e}(t)$ as $\widehat{M}\,(\ddot{\boldsymbol{e}} + \,K_1\,\dot{\boldsymbol{e}}\, +\,K_0\,\boldsymbol{e}) = \,\widetilde{M}\ddot{\boldsymbol{q}} + \,\widetilde{\boldsymbol{h}}$ where $\widetilde{M} = \widehat{M} - M$ and $\widetilde{\boldsymbol{h}} = \widehat{\boldsymbol{h}} - \boldsymbol{h}$.
4. Rewrite the right side of the error dynamics, that is $\widetilde{M}\ddot{\boldsymbol{q}} + \,\widetilde{\boldsymbol{h}}$, in the linearly parameterized form $Y(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\,\widetilde{\boldsymbol{p}}$ where $\widetilde{\boldsymbol{p}} = \widehat{\boldsymbol{p}} - \boldsymbol{p}$ is the difference between the parameter vector $\boldsymbol{p}$ and its estimate $\widehat{\boldsymbol{p}}$.
5. Using the previous step, rewrite the error dynamics as $\ddot{\boldsymbol{e}} + \,K_1\,\dot{\boldsymbol{e}}\, +\,K_0\,\boldsymbol{e} = \Phi\,\widetilde{\boldsymbol{p}}$ where $\Phi = \widehat{M}^{-1}Y(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$. Under what condition can Φ be defined?
6. Choose K_0 and K_1 such that the error dynamics is stable.

7. Rewrite the error dynamics $\ddot{\boldsymbol{e}} + K_1\, \dot{\boldsymbol{e}} + K_0\, \boldsymbol{e} = \Phi\, \widetilde{\boldsymbol{p}}$ in the state space form $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\Phi\widetilde{\boldsymbol{p}}$ where $\boldsymbol{x} = [\boldsymbol{e}\ \dot{\boldsymbol{e}}]^{\boldsymbol{T}}$. Specify the matrices A and B. Why is matrix A stable?
8. Knowing that matrix A is stable, choose a symmetric positive definite matrix Q and solve the Lyapunov equation $A^T P + PA + Q = 0$, manually or in MATLAB, to find the symmetric positive definite matrix P. Which result, theorem or lemma, in the book supports this step?
9. Choose a symmetric positive definite matrix Γ to form the Lyapunov function candidate $V(\boldsymbol{x}) = \boldsymbol{x}^T P\boldsymbol{x} + \widetilde{\boldsymbol{p}}^T \Gamma \widetilde{\boldsymbol{p}}$. Show that $\dot{V}(\boldsymbol{x}) = -\boldsymbol{x}^T Q\boldsymbol{x} + 2\widetilde{\boldsymbol{p}}^T [\Phi^T B^T P\boldsymbol{x} + \Gamma\, \dot{\widetilde{\boldsymbol{p}}}]$. Justify, using the corresponding theorem(s) in the book, why the parameter estimation dynamics $\dot{\widehat{\boldsymbol{p}}} = -\Gamma^{-1}\Phi^T B^T P\boldsymbol{x}$ makes the system stable.

Simulation Model

Make a Simulink model for the abovementioned designed system subject to the following requirements. Use the template file *AdaptivelyControlledPendulum.slx* for this purpose.

1. The top layer of your Simulink model consists of three following subsystems only (the bolded words should be the names of the subsystems).
    a. **Reference Signals** subsystem
    b. **Controller** subsystem
    c. **Pendulum** (plant) subsystem
2. The required signals are logged and shown through *Data Inspector* (no oscilloscopes). All the logged (and shown) signals should bear meaningful names.
3. Within each subsystem you may use different blocks. It is recommended to use also MATLAB subsystems blocks to implement more difficult mathematical functions by code.
4. The model parameters can be entered in the model but not in *MATLAB Base Workspace*.
5. In the **Reference Signals** subsystem you may use different blocks to generate the reference signal $\boldsymbol{q}^d$ and its derivatives $\dot{\boldsymbol{q}}^d$ and $\ddot{\boldsymbol{q}}^d$.
    a. Make sure that you generate reference signals through integration (that is, calculate $\boldsymbol{q}^d$ from $\dot{\boldsymbol{q}}^d$, and $\dot{\boldsymbol{q}}^d$ from $\ddot{\boldsymbol{q}}^d$) not derivative blocks (that is, the other way around).
    b. See the following section for the types of reference signals.

Simulation Scenarios And Analysis

You run two simulations and analyze them as explained below. Note that for parameter estimation you need to use an initial value (estimate). Use $8\ [kgm^2]$ and $5\ [Nm]$ as the initial values of $\hat{I}$ and $\widehat{mgl}$, respectively.

1. Unit-Step Response simulation:
    a. In the subsystem block *Reference Profile* of *AdaptivelyControlledPendulum.slx* see the MATLAB Function block *Acceleration Profile Generator*. This block provides the acceleration profile of the required simulation. You can check the MATLAB function to see the formulas and how this profile has been made. The *Stop Time* of the Simulink file has been set to 2 seconds, which is the required time of the simulation.
    b. Complete the *Reference Profile* through making the reference signals for velocity and position, which is the main reference for the controller.
    c. Run the simulation. Use the default *Variable-Step* solver for this purpose.

  i. Log the signals $\theta, \dot{\theta}, \ddot{\theta}$ and their references, and also the *controlTorque* and parameter estimates in Data Inspector. Plot the signals $\theta, \dot{\theta}, \ddot{\theta}$ and its corresponding reference on the same figure to make analysis easier.
  ii. Comment on the motion profile signals through comparing them with the standard signals (step, ramp and parabola), and the initial, intermediate and final values of the signals during the profile.
  iii. Comment on the convergence of the pendulum behavior to the reference signals. Justify what you see using the theory you used during the design.
  iv. Comment on the convergence of the parameters to their final values, too. Justify what you see using the theory you used during the design.
  v. Comment on the *controlTorque* and the ratings of its required actuator, say an electric motor.

 d. Run the simulation this time by changing the *Variable-Step* solver to a *Fixed-Step* solver with a step-size which results in the graphs which are as close to the previous case (with *Variable-Step* solver). Show the resulting graphs in Data Inspector, too. This step helps to figure our at which sampling time you can implement your model in Home-Taken Exam 1c for Real-Time Systems.
  i. NOTE: You may get an idea of the used step-size of the Variable Solver by checking the *Properties* window (bottom left) of the *Data Inspector* attached to the *Variable-Step* Run.

2. Sinusoidal Response simulation
 a. In the subsystem block *Reference Profile* of *AdaptivelyControlledPendulum.slx*, replace the MATLAB Function block *Acceleration Profile Generator* with a Sinusoidal signal generator, for example using Signal Generator block. You shall do two simulations, both using 2 $[m/s^2]$ for the magnitude of the sinusoid. For the frequency of the sinusoid use 2 $[Hz]$ for the first one, and 3 $[Hz]$ for the second simulation. Set the *Stop Time* for both simulations to 10 seconds.
 b. Make sure that the *Reference Profile* is complete through making the reference signals for velocity and position, which is the main reference for the controller.
 c. Run the two simulations.
  i. For each simulation, log the signals $\theta, \dot{\theta}, \ddot{\theta}$ and their references, and also the *controlTorque* and parameter estimates in Data Inspector. Plot the signals $\theta, \dot{\theta}, \ddot{\theta}$ and its corresponding reference on the same figure to make analysis easier.
  ii. Comment on the convergence of the pendulum behavior to the reference signals. Justify what you see using the theory you used during the design.
  iii. Comment on the convergence of the parameters to their final values. Compare the rate of parameter convergence for two simulations, and justify what you see using what you learned for the case of linear systems in the textbook [1].

References

[1] J. J. E. S. a. W. Li, Applied nonlinear control (Vol. 199, No. 1, p. 705), Englewood Cliffs, NJ: Prentice hall., 1991.

[2] J. D. P. A. E.-N. G. F. Franklin, Feedback Control of Dynamic Systems, Pearson Education, 2020.