**MAE 282** Prof. M. Krstic

Adaptive Control Fall 2006

# FINAL EXAM

---

- Take home.
- No collaboration.
- **Do one problem from the list.**
- Besides writing a "report" with your results, I want every student to prepare 2-3 slides in Power Point which you will present to me and rest of the class.
- The oral presentation of your slides is scheduled at 7pm on Monday, December 4th (this is the official scheduled time for our final exam). Location: my lab, room 2101 EBU1.
- If anyone absolutely cannot make the oral presentation on December 4th, the make up opportunity will be at 5pm on Thursday, December 7th, same location.
- You bring your computers, I provide a projector.

---

**Problem 1.** Consider the plant

$$G(s) = g\frac{s-b}{(s^2+p)^2+q}\,,$$

where the parameters are all positive and unknown.

1. Design an identifier based on a gradient update law.
2. Design a pole placement controller for tracking a sinusoid of frequency 1 rad/sec.
3. Simulate the plant with the adaptive controller for

   $$g = p = q = 1, \quad b = 2\,.$$

   Make sure to choose the initial conditions for the parameter estimator that give destabilizing initial control gains. (Set the adaptation gain to zero and try a few initial conditions for the parameter estimates until you find some for which the plant output is growing unbounded. Then turn on the adaptation for the subsequent simulations.) Are you achieving stabilization and tracking? Are your parameter estimates converging?

4. Add some measurement noise at the plant output. This will make your parameter estimator prone to drifting. Apply a deadzone in the update law. Tune its threshold to prevent the drifting.

   (The deadzone is explained in Section 8.4.3 of Ioannou and Sun's book. Use the continuous deadzone in Figure 8.5(b). If you are more comfortable with equations than with pictures, you can use equation (8.4.33). However, you don't really need this equation because you can use a deadzone as a block in Simulink. Note also that the deadzone should be applied only to the estimation error part of the update law. Everything else remains unchanged, namely, the regressor, the normalization, and the adaptation gain. Unlike what is shown in Section 8.4.3, you should use the normalization. Some tips on the use of deadzone: If the deadzone threshold $g_0$ is too high, your performance in the absence of noise will suffer, perhaps even the close loop stability will suffer; if the threshold $g_0$ is too low, you will have parameter drift; the best thing is to start with a low $g_0$ and keep increasing it until the drift is eliminated but before the stability starts to suffer; obviously, if the noise is very large, you will never be able to even achieve stability because you will need a huge $g_0$.)

**Problem 2.** Starting with the notes distributed in class on PID tuning using extremum seeking and using the Matlab code downloadable from

`http://flyingv.ucsd.edu/krstic/teaching/282/zemler.zip`

conduct a case study of stabilizability and performance tuning with PID control for various classes of single-input-single-output plants. Go beyond the examples $G_1(s), G_2(s), G_3(s), G_4(s)$ and consider plants that contain other "difficulties," like one unstable pole, resonant pairs of poles, in addition to the difficulties dealt with in the notes like pure delay and unstable zeros. You can consider also the presence of noise and/or saturation if you like, but the focus should be on stabilization and performance of *linear* plants.

**Problem 3. ("The Carlos Cox problem")** This problem deals with true *real-time* control parameter tuning via extremum seeking. However, to simplify things, we don't consider the class of PID controllers but a class of particularly simple "on-off" (or "bang-bang," if you wish) controllers which are often used in practice. In some applications in industry one encounters problems where the actuation authority is low, so one wants to use the full 'control power.' For example, in control of combustion instabilities, one may have to use "on-off" valves to control the process. With such an actuator, one has control only of the timing of the control inputs and not of their size. So it is reasonable to use control laws of the form $u(t) = \text{sgn}(y(t-\theta))$, where $\theta$ is a delay time, or, in the frequency domain, with some abuse of notation, $u(s) = \text{sgn}\left(\text{e}^{-s\theta}y(s)\right)$. You can also define the control law with the opposite sign, namely, $u(t) = -\text{sgn}(y(t-\theta))$. In some applications this sign difference will be important, whereas in other applications it won't be so important. For example, in applications where the plant response is oscillatory, namely, where $y(t)$ tends to be sinusoidal, the sign difference between the two variants of the control law can be accounted for using an additional delay of $\pi$ in *theta*.

So, for applications discussed above, it is reasonable to consider using ES to tune $\theta$ to some optimal value. The figure on the last page shows an extremum seeking scheme that should be able to do that. In the lower branch of the feedback loop, on the left, you will find the standard ingredients of ES—the probing signal, the integrator, the demodulation, and a washout filter—whereas on the right of the lower branch you will see one way to calculate a reasonable cost

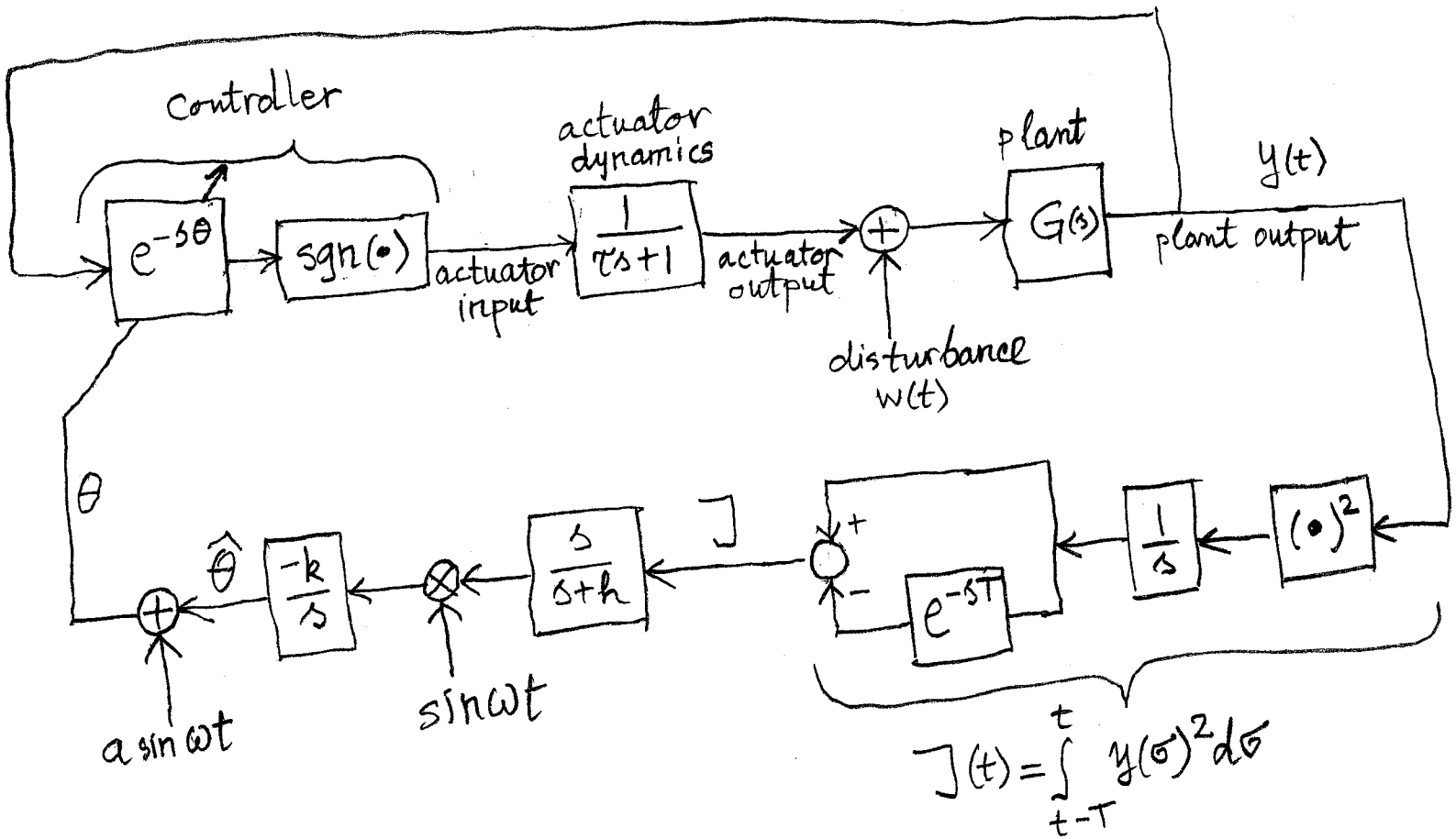
functional. This cost functional is given as an integral of the square of the output over a moving window of length $T$, namely, $J(t) = \int_{t-T}^{t} y(\sigma)^2 d\sigma$. The block diagram shows a particularly simple way to calculate such a functional in real time using basic Simulink blocks for an integrator, a delay, and a square function.

Note that the objective in this application is not to optimize step responses but to continuously monitor the regulation ability of the controller and adjust $\theta$ to minimize $J$. To make the problem realistic (this is how it is in the case of combustion instabilities), I have added a disturbance $w(t)$ in the block diagram. The presence of a disturbance, particularly if it is stochastic white noise, will make it impossible for the control to ever settle. However, in reality this will not be an issue. The actuator will have its own inertial (its time constant $\tau$) and it will not respond to "chattering" waveforms of the actuator input (usually some voltage). If the actuator is a valve, it will only partly open when driven by a "chattering" voltage waveform. To model all of this, I have inserted the "actuator dynamics" block $\frac{1}{\tau s+1}$ in the regulation loop, right after the "controller."

Note that the controller $u(t) = \text{sgn}(y(t-\theta))$ is a very "low-tech" controller and that it will not work great for many plants. But this is something that is often used in industry under the motivation of exploiting the "full actuator range," so it is worth learning one way how to tune such a controller in real time.

Like Problem 2, this is an open-ended problem. You have many choices left up to you, including the choice of the plant $G(s)$, the actuator time constant $\tau$, all of the parameters of the extremum seeking scheme, and even the choice whether you will use positive gain, $u(t) = \text{sgn}(y(t-\theta))$, or negative gain, $u(t) = -\text{sgn}(y(t-\theta))$, noting of course that this last choice may not be important because with this class of controllers we achieve things with timing, not with the size of the control input.

Explore the capabilities of this scheme for several plant choices. This kind of controllers have been used in combustion instabilities where $G(s)$ contains a lightly damped pair of stable poles and possibly some delay.


controller
actuator dynamics
plant
y(t)
$e^{-s\theta}$
sgn(•)
actuator input
$\frac{1}{\tau s+1}$
actuator output
G(s)
plant output
disturbance w(t)
θ
$\hat{\theta}$
$\frac{-k}{s}$
$\frac{s}{s+h}$
$\int$
+
−
$e^{-sT}$
$\frac{1}{s}$
$(\bullet)^2$
a sin ωt
sinωt
$\int(t) = \int_{t-T}^{t} y(\sigma)^2 d\sigma$