18 Adaptive Control

Adaptive control is the attempt to “redesign” the controller while online, by looking at its performance and changing its dynamic in an automatic way. It was motivated as an extension of gain scheduling for aircraft autopilot design, allowing the system to account for previously unknown dynamics. Adaptive control is a feedback law that observes the process and the performance of the controller and reshapes the controller closed loop behavior autonomously.

Adaptive Control is made by combining
1. online parameter estimator based on current measurements and control actions
2. a control law, that recalculates the controller based on the estimated parameters

Example 1
Consider the linear system with unknown gain:
\[
\dot{x} = ax + bu,
\]
Where \(a, b\) are unknown, but positive so that the system is unstable, and the system has positive gain. Then the following adaptive control law will stabilize the system:
\[
\dot{k} = x^2 \\
\dot{u} = -kx
\]
As \(x\) grows, the gain \(k\) grows, until it is large enough, that is \(a-bk<0\). The system will then stabilize.

In general the sign of \(b\) need not be known. Here the so-called Nussbaum controller will stabilize the system:
More generally, by exploiting generalized switching functions similar to the one here, multiple-input multiple-output systems with unknown gains may be stabilized. The disadvantage is that there is no guarantee of how long it takes to find the stabilizing controller, and no bound on the transients that may be experienced before it is found. For this reason, such “universal” adaptive controllers are not so interesting, and other methods are preferred.

There are two main types of adaptive control. We discuss them below.

18.1 Indirect case:

In this case, the parameters which are estimated are those of the plant, and the controller is designed based on those parameters. For example, an energy consumption coefficient is estimated; a linearised version of the plant is produced, which is then used in a pole placement method to calculate the gains of the final controller. The main problem is to make sure that the controller behaves well in cases when the estimation of the plant is not good, like for example, during transients. In principle, the indirect method is applicable to all types of plants.
18.2 Direct case

In this case the controller parameters are estimated directly without intermediate steps. The controller in Example 1 above is such a controller. Another example is the self tuning PID loop. The same problems as in the indirect case can be encountered here. This method can be safely applied only to minimum phase systems.

A specific adaptive control architecture which is often used is \textit{MRAC – Model Reference Adaptive Control}. In this case the control objective is to tune the controller so that the closed loop dynamics mirror those of a reference system which is represented by an ideal model. An example could be when a car tunes the steering and engine characteristics so that the car handles like it was on a dry road, although it is actually on a wet, icy or unsealed road. In this case the control architecture is as follows:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig4.png}
\caption{Example MRAC Architecture}
\end{figure}
18.3 Design of Online Parameter Estimator

Parameter estimation is at the very core of adaptive control, as is particularly obvious in the indirect method. When making a more complete study of adaptive control, the proof of convergence of the system including the estimation and controller parameterization is one of the key technical difficulties. Techniques which are commonly applied to prove convergence include

1. Lyapunov / passivity techniques
2. Averaging theory

Parameter estimators may be constructed by a variety of methods. In the following subsection we provide some specific examples.

18.3.1 Sensitivity method:

This method is related to the sensitivity equation, where the parameter is estimated by using a nominal trajectory and its sensitivity with respect to that parameter

\[ x(t, \lambda) - x(t, \lambda_0) = S(t)(\lambda - \lambda_0) \Rightarrow \dot{\lambda} = \lambda_0 + S(t)^{-1}(x(t, \lambda) - x(t, \lambda_0)) \]

As the number of states and the number of parameters are in general not the same, the pseudo inverse (least squares algorithm) of the sensitivity matrix must be calculated, and in most cases the matrix will not be invertible.

18.3.2 Gradient method

This method is driven by the idea of minimizing the square of the prediction error.

\[ J(K) = 0.5e^2(K) \Rightarrow \dot{\lambda} = -\gamma \frac{\partial e}{\partial \lambda} \]

Since the parameters are unknown, the main problem is the calculation of the partial derivative in real time. One well known approach to address this is the so called MIT rule.

The basic set up is shown in below

![Figure 5. MIT rule setup](image-url)
The plant is $k_p Z_p(s)$, where $k_p$ is unknown apart from its sign, and $Z_p(s)$ is a known stable transfer function. The basis of identification is that a adjustable positive known gain $k_c(t)$ is introduced as shown, and the output of the upper branch is compared to the output of $k_m Z_p(s)$ when the same driving signal is applied; where $k_m$ is a known gain.

The MIT rule is the rule of adjustment for $k_c(t)$. The idea is to use gradient descendent to adjust $k_c(t)$, which in this case leads to

$$
\dot{k}_c = -g(y_p - y_m)y_m
$$

The rule is appealing due to its simplicity but it often leads to instabilities. The mechanism is one where the fast dynamics of the adaptation destabilizes the closed loop system (based on the controller that used the estimated parameter). In applications, it is important to understand which are the time constants in the system to avoid self inflicted oscillations.

### 18.3.3 Extended Kalman Filter

The so called Extended Kalman Filter is a related alternative. Kalman filters are normally used for state estimation. However, using a device called state augmentation that converts the parameters into states with no dynamics but with known process noise covariance it becomes possible to apply this technique to this case. See below for details

\[
\begin{align*}
\dot{x} &= f(x, u, p) + \omega_1 \\
\dot{p} &= 0 + \omega_2 \\
y &= h(x, p) + \eta
\end{align*}
\]

\[
\begin{align*}
E(\omega_1 \omega_1) &= Q \\
E(\omega_1 \omega_2) &= Q_{pp} \\
E(\eta \eta) &= R
\end{align*}
\]
When does this algorithm work?
1. No comprehensive theory.
2. Equivalent to the controllability of a nonlinear system by output feedback
3. Detectability” and “stabilizability” of the linearizations are necessary but not sufficient conditions
4. Implicity applies the separation principle – the principle that any stabilizing controller may be separated into a state estimator and state feedback operator.

Is the data required always available?
1. Main problem is selecting Q, Qpp and R. They are “tuning” parameters.
2. Some adaptive (self-tuning) methods treat the system as a nonlinear plant to be “stabilized” via Q and R.

Higher order filters are used when strong nonlinearities render EKF useless

18.4 Observability and Persistency of Excitation, Bursting Phenomena.

One must be aware of the fact that trying to identify more parameters than what the information available can deliver may also lead to instabilities. Obviously, the parameters must be observable – or estimateable. Additionally the system must be “excited” so as to allow estimation of the parameters. If certain modes of behaviour are not excited, the corresponding parameters may not be estimated (e.g. Neural Networks). Additionally the
update of the sensitivity matrix in the recursive estimation procedure may lead to an over-
sensitivity of the algorithm to excitation of previously unexcited states, which in turn causes erroneous updates of the parameters.

For example, there exist a class of phenomena known as bursting phenomena, see the figure below. Sometimes, instability is observed when the signals become steady. After a furious phase, the system goes back to steady state again.

![Figure 7. Bursting](image)

This reason behind this behavior is that when the signals are near zero the adaptive algorithm does not have enough information in order to identify all parameters simultaneously. This leads to divergence of the estimates. The consequence of that is a faulty controller (because it is based on those parameters) rendering the plant unstable. Once the signals become rich enough though, the adaptive algorithm begins to deliver good estimates of the parameters. Now, just in time, the controller begins to perform well, leading to plant recovery. Eventually, the problem appears again.

Clearly, in industrial applications bursting must not occur. There are several possibilities to avoid problems related to bursting or instabilities in the estimation, for example:

1. Inject excitation signals into the control loop
2. Introduce a Dead-Zone – adaption only occurs when errors are large enough
3. Projection in the parameter space – constrain the domain in which the parameters are allowed to lie
4. Dynamic Normalization – slow the adaption relative to the rate of model error growth.