"Automatic Stabilization of an Unmodeled Dynamical System" Final Report

Gregory L. Plett and Clinton Eads

May 2000.

1 Introduction

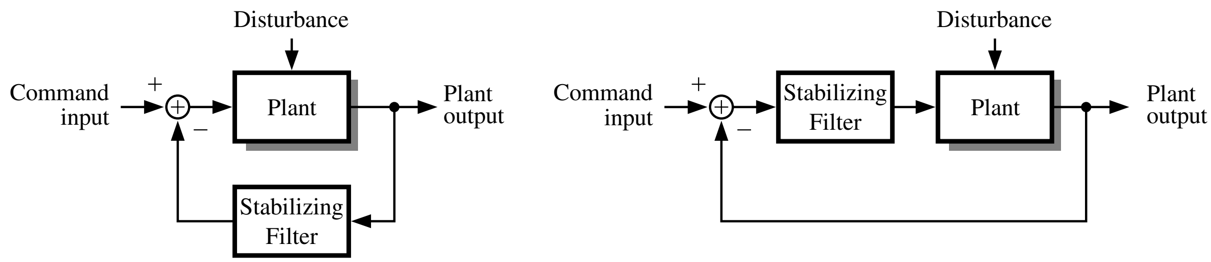
Automatic control systems are commonly encountered in daily life. An automobile cruise-control system regulates the speed of a vehicle via measuring the current speed and computing the correct throttle position. An aircraft autopilot can perform all the operations necessary to take off, navigate a flight-path and land an airplane without human intervention.

The field of control theory concerns itself with the analysis and design of such automatic control systems. The dynamical system to be controlled (*e.g.*, the automobile or aircraft) is called the "plant" in control systems literature. The plant need not be a mechanical system: electrical, electro-mechanical, thermo-dynamic and fluid-dynamic systems are other classes of systems which may be controlled with an appropriate controller.

The goal of a control-system is to make the plant output behave in a user-specified manner as accurately and robustly as possible. For example, the automobile cruise-control system must maintain a vehicle's speed at a constant level. If the vehicle is moving too slowly, then the throttle must be opened; if the vehicle is moving too quickly, the throttle must be closed. The cruise-control system must also be able to compensate for external disturbances: *e.g.*, wind gusts and hills and valleys in the road.

One important aspect of control-system design regards *stability*. A control system is said to be stable (mathematically) if the output of the system has finite (bounded) values for any finite (bounded) input. In less precise but more practical terms, a system is stable if its output behaves in a "reasonable" way for any reasonable input. Thus, insuring stability is a necessary requirement placed on any control-system design. Once the system is stable, it may be fine-tuned to improve performance even further.

There are many important control-design applications for unstable and potentially-unstable systems (*e.g.*, feedback audio amplifiers, and advanced fighter aircraft). In order to design a controller to stabilize these systems, the standard techniques require that the designer must first develop a mathematical model of the system to be stabilized. This model is determined using Newton's laws for mechanical systems, Kirchoff's laws for electrical systems, and so forth. The model typically contains certain parameters (such as



(a) METHOD I (b) METHOD II

Figure 1: Conventional methods to stabilize a plant.

mass values, resistances, capacitances, etc.) which must be determined experimentally. This model is then used by a control-systems engineer to design an appropriate stabilizing system.

Deriving the mathematical model and precisely determining the values of all the parameters of the model is a very difficult and time-consuming task for an unstable system. This report addresses research performed under the 1999–00 CRCW award titled "Automatic Stabilization of Unmodeled Dynamical Systems" to develop methods to automatically stabilize an unstable system *without* needing to model the system or determine its parameters first.

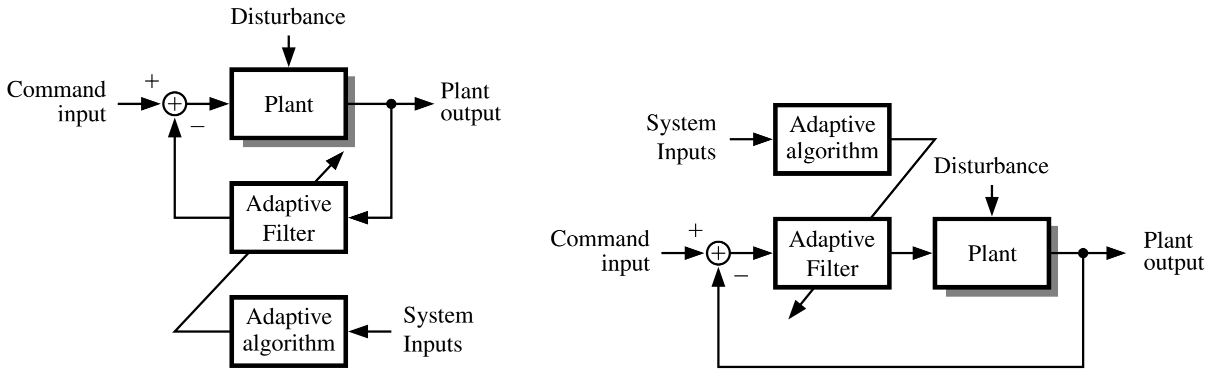
2 Objectives and Methodology

If a model of the plant has been derived and all of its parameters are known, one of two conventional methods is used to stabilize the system as shown in Fig. 1. In Fig. 1(a), the output of the plant is "fed back" to the input through a specially-designed circuit called a "filter." The control-systems engineer designs the feedback filter in such a way that it stabilizes the overall system. In Fig. 1(b), the filter is placed in a different configuration, allowing different stabilizing properties.

In this work, we addressed the problem of stabilizing an unstable system for which no mathematical model has been derived, and for which no parameters are known. Stabilization of such a system is a difficult problem which has not yet been solved. We investigated methodologies in keeping with adaptive inverse control [1, 4, 6]. An adaptive filter was used in a feedback configuration in order to stabilize the plant. Simplified schematic representations of this concept are shown in Fig. 2. These are the same as Fig. 1, except that the stabilizing filter is no longer designed by a control-systems engineer. An algorithm adapts the operation performed by the filter (based on data collected while the system operates) in order to stabilize the overall system.

3 Adaptive Filters

A filter is a device which performs a mathematical operation on its input signal in order to produce an output signal. Generally, a filter computes a function based on its current and past input values, and past output



(a) METHOD I (b) METHOD II

Figure 2: Two approaches to an adaptive stabilizing system.

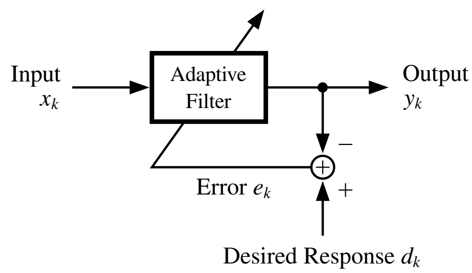


Figure 3: An adaptive filter.

values. For example,

$$y_k = fn(x_k, x_{k-1}, \ldots, x_{k-m}, y_{k-1}, y_{k-2}, \ldots, y_{k-n}).$$

Linear filters were used in this work. A linear filter computes its output as a weighted sum of input and output values.

$$y_k = \sum_{i=0}^{m} b_i x_{k-i} + \sum_{i=1}^{n} a_i y_{k-i}.$$

An adaptive filter is a filter with an additional "special" input, as depicted in Fig. 3. This input is the "desired response," and specifies exactly what the output of the filter should be at each time instant. An algorithm such as LMS [3, 5] is then used to adapt (modify) the filter parameters (*e.g.,* the a_i and b_i values of the linear filter) in order for the filter output to more exactly resemble the desired output. This adaptation is based on an error signal $e_k = d_k - y_k$ where d_k is the desired-response signal. Generally, the challenge when using adaptive filters is coming up with a d_k signal suitable to adapt the filter. Two different methods were attempted here.

METHOD I: The first method considered using a stabilizing architecture as in Fig 1(a). The block labeled as the plant will henceforth be referred to as the transfer function matrix $[P(z)]$.[1] The block labeled as the stabilizing filter will be referred to by its transfer function matrix $[R(z)]$. We require a method of obtaining a desired-response signal in order to use an adaptive algorithm such as LMS to adapt $[R(z)]$ to stabilize the system.

We approached this problem by designing a transfer function $[H(z)]$ which specifies the desired transfer function of the feedback system formed by $[P(z)]$ and $[R(z)]$. The feedback system has transfer function $[P(z)]\big[I + [R(z)][P(z)]\big]^{-1}$, and so the goal is to adapt $[R(z)]$ such that

$$[P(z)]\big[I + [R(z)][P(z)]\big]^{-1} \to [H(z)]$$

without knowing $[P(z)]$.

To do so, we imagine an $[R^*(z)]$ filter which is able to compute the correct function. We find $[R^*(z)]$ as

$$\begin{aligned} [P(z)]\big[I + [R^*(z)][P(z)]\big]^{-1} &= [H(z)] \\ [P(z)] &= [H(z)]\big[I + [R^*(z)][P(z)]\big] \\ [P(z)] - [H(z)] &= [H(z)][R^*(z)][P(z)], \end{aligned}$$

assuming that $[H(z)]$ is invertible (and is stable when inverted), then

$$[H(z)]^{-1}\big[[P(z)] - [H(z)]\big] = [R^*(z)][P(z)]$$

or

$$[H(z)]^{-1}[P(z)] - I = [R^*(z)][P(z)].$$

This equality holds for $[R(z)] = [R^*(z)]$, the optimal choice of the $[R(z)]$ filter. Any inequality between these two sides is due to errors in $[R(z)]$, and these errors may be used to adapt $[R(z)]$ to approach $[R^*(z)]$. In block-diagram form, this would appear as in Fig. 4. Note that it may not be possible for $[R(z)]$ to be adapted to equal $[R^*(z)]$ due to constraints on the architecture of $[R(z)]$. For example, $[R(z)]$ is constrained to be causal, and may additionally be constrained to be FIR.

If $[H(z)]$ incorporates a pure-delay term, then it cannot be inverted with a causal inverse. However, since we design $[H(z)]$, we are free to design one as $[H(z)] = z^{-k}[H_{mp}(z)]$, where z^{-k} denotes a delay of k samples, and $[H_{mp}(z)]$ is "minimum phase," having a causal and stable inverse. Then, we have

$$[P(z)] - z^{-k}[H_{mp}(z)] = z^{-k}[H_{mp}(z)][R^*(z)][P(z)]$$

[1]For this work, we will assume that all subsystems are linear, time invariant (LTI) and so have transfer functions. Furthermore, we allow multi-input-multi-output (MIMO) systems, and so the transfer function is in actuality a transfer function matrix. Because matrix multiplications are not commutative, we will take special care to write equations in the correct order for MIMO systems. For a single-input-single-output (SISO) system, the derivation can often be simplified.

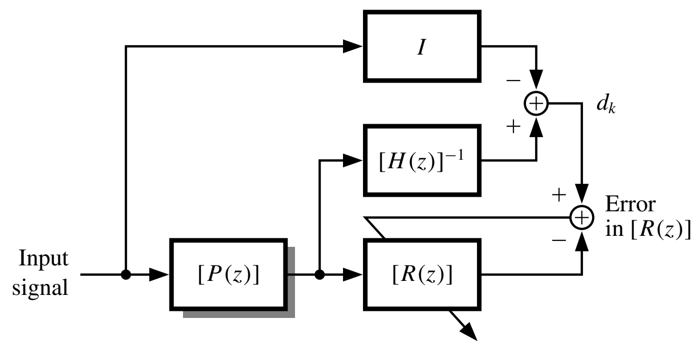


Figure 4: First method to adapt $[R(z)]$.

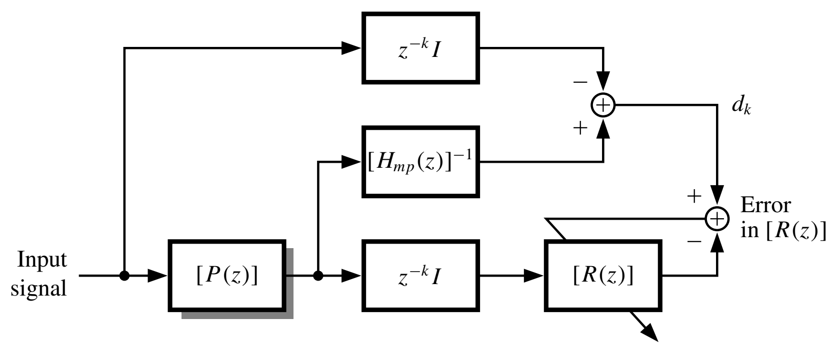


Figure 5: Second method to adapt $[R(z)]$.

$$[H_{mp}(z)]^{-1}[[P(z)] - z^{-k}[H_{mp}(z)]] \quad = \quad z^{-k}[R^*(z)][P(z)],$$

or

$$[H_{mp}(z)]^{-1}[P(z)] - z^{-k}I = z^{-k}[R^*(z)][P(z)].$$

The block diagram showing how to adapt $[R(z)]$ in correspondence to this equation is drawn in Fig. 5.

From these diagrams, it appears that we still need to know the dynamics of $[P(z)]$ in order to adapt $[R(z)]$. This is not the case—we use the actual plant in-place to compute the output of the $[P(z)]$ block. The complete block-diagram of the system is shown in Fig. 6. In the figure, $[R_{copy}(z)]$ is a filter with identical coefficients (*e.g.,* a_i and b_i) to $[R(z)]$, but has different inputs at any point in time. (Adapting the coefficients of $[R(z)]$ instantly adapts the coefficients of $[R_{copy}(z)]$ as well).

METHOD II: The second stabilizing method uses the standard stabilizing architecture in Fig 1(b), but with an adaptive stabilizing filter. This system has transfer function

$$[P(z)][R(z)]\big[I + [P(z)][R(z)]\big]^{-1}.$$

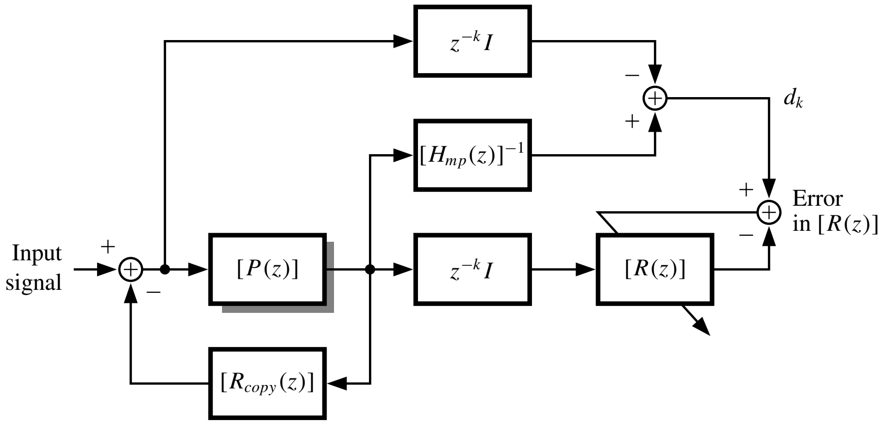


Figure 6: Final "Method I" for adapting $[R(z)]$ in place.

As before, we equate this transfer function to $[H(z)]$, and solve for $[R^*(z)]$.

$$\begin{aligned}
[P(z)][R^*(z)]\big[I+[P(z)][R^*(z)]\big]^{-1} &= [H(z)] \\
[P(z)][R^*(z)] &= [H(z)]\big[I+[P(z)][R^*(z)]\big] \\
\big[I-[H(z)]\big][P(z)][R^*(z)] &= [H(z)] \\
[P(z)][R^*(z)] &= \big[I-[H(z)]\big]^{-1}[H(z)] \\
[R^*(z)]^{-1}[P(z)]^{-1} &= [H(z)]^{-1}\big[I-[H(z)]\big] \\
I &= [R^*(z)]\big[[H(z)]^{-1}-I\big][P(z)].
\end{aligned}$$

This final line is suitable for adapting $[R(z)]$. Also, if $[H(z)] = z^{-k}[H_{mp}(z)]$, where $[H_{mp}(z)]$ has a stable and causal inverse, the equation becomes

$$z^{-k}I = [R^*(z)]\big[[H_{mp}(z)]^{-1} - z^{-k}I\big][P(z)].$$

A block-diagram for adapting $[R(z)]$ using this method is shown in Fig. 7

4 Results

Simulations were performed to test the two stabilization methods. After initial success with simple unstable systems, we discovered systems for which neither method worked. As an example, consider the simple plant

$$P(z) = \frac{1}{z^2 + 2.4z + 1.35}$$

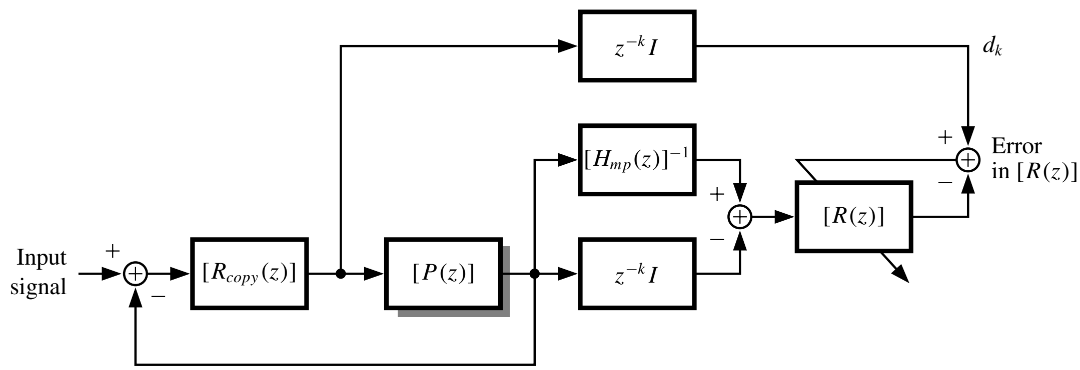


Figure 7: Final "Method II" for adapting $[R(z)]$ in place.

which has poles at -1.5 (unstable) and -0.9 (stable). Our desired transfer function for the system was chosen to be

$$H(z) = \frac{1}{z^2 + z + 0.25}$$

which has both of its poles at -0.5 (stable). Using method I, we can solve analytically for the required $R(z)$ to achieve this transfer function, and we find that

$$R(z) = -1.4z - 1.1$$

which is non-causal! This appears to be a fundamental problem with method number I. If we have knowledge of the plant, we can find a range of acceptable pole locations for $H(z)$ which yields a stable and causal $R(z)$. For this example, suppose $R(z) = k_0 + k_1 z^{-1}$, then k_0 and k_1 must be within the banana-shaped region in Figure 8 to stabilize the system. This is a very small range of acceptable parameters, and precise knowledge of $P(z)$ is required in order to achieve this solution. This goes against the premise of the project.

Using method II and solving for $R(z)$ we find

$$R(z) = \frac{z^2 + 2.4z + 1.35}{z^2 + z - 0.75}.$$

This is a causal solution, but it is unstable. It also attempts to perform an unstable pole-zero cancellation which is not permissible. Further research discovered the method of Ragazzini [2] which is a non-adaptive method to design controllers which is very similar to method II. It turns out that there are restrictions on allowable $H(z)$ which depend on knowing the plant dynamics $P(z)$. Again, this goes against the premise of the project.

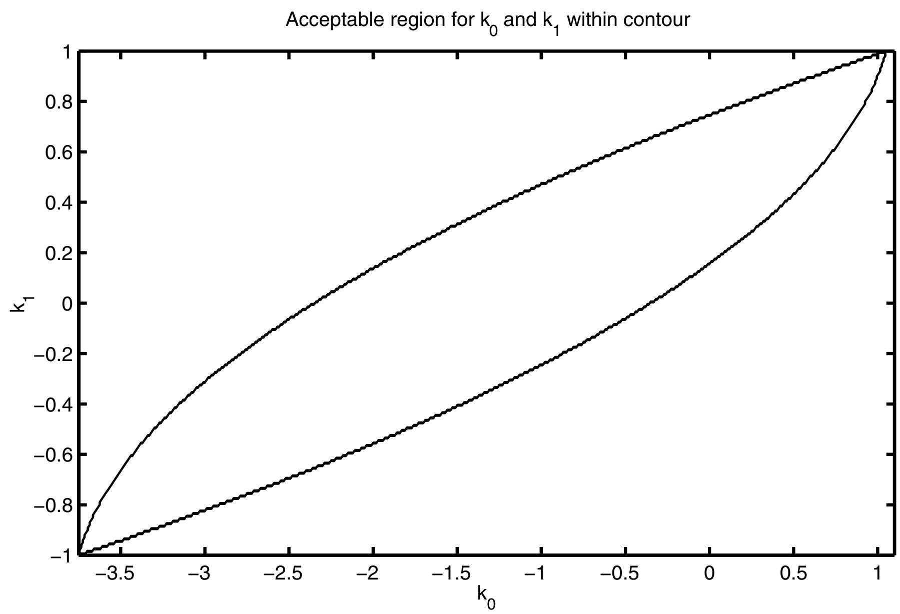


Figure 8: Valid k_0 and k_1 region.

5 Conclusions

We regret that neither of the two methods attempted were able to achieve the goals of this project. We discovered that both methods require explicit knowledge of the plant dynamics in order to design an appropriate $H(z)$ to stabilize the system. Our goal was to stabilize the system without knowing the plant dynamics. We hope to develop functional methods in the future, but these two ideas did not work.

References

[1] M. Bilello. *Nonlinear Adaptive Inverse Control*. PhD thesis, Stanford University, Stanford, CA, April 1996.

[2] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, Reading, MA, third edition, 1990.

[3] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, NJ, third edition, 1996.

[4] G. L. Plett. *Adaptive Inverse Control of Plants with Disturbances*. PhD thesis, Stanford University, Stanford, CA 94305, May 1998.

[5] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[6] B. Widrow and E. Walach. *Adaptive Inverse Control*. Prentice Hall P T R, Upper Saddle River, NJ, 1996.