AN ADAPTIVE VIEW OF SYNCHRONIZATION

William A Sethares

Dept Electrical Engineering
University of Wisconsin
Madison WI 53706

John M. Walsh, and C. R.Johnson, Jr.

School of Electrical Engineering
Cornell University
Ithaca NY 14853

ABSTRACT

There are four basic synchronization tasks in communications systems: the receiver must determine the phase and frequency of the carrier signal, and it must determine the timing and period of the symbol clock. All four of these tasks can be solved via gradient methods, and many common algorithms such as the Phase locked Loop, Costas Loop, Band-edge timing, and output energy maximization (for clock recovery) can be viewed as a gradient algorithm applied to specific cost functions. This paper presents an overview of synchronization by stating a formal definition of an "adaptive element". By looking at the cost functions, insight can be gained as to how the varous elements can be combined, sequenced, and compared.

Introduction

This paper defines an abstract "adaptive element" as a parameterized subsystem with input and output signals, a system function (that specifies how the output is constructed from the input), a cost function (that helps determine the "best" value for the parameters), and a stepsize (that specifies how fast the parameter may change). The cost function can be used to draw the error surface, which is useful in understanding gross features of the element such as how many fixed points it may have. The "sensitivity" function of the adaptive element (the derivative of the cost with respect to a parameter of interest) is useful as a measure of the responsiveness of the element to changes in the parameters. Speaking loosely, higher sensitivity implies faster convergence of the adaptation, though it may also be associated with undesirable aspects such as suseptibility to noise. The sensitivity function may be useful as a tool to choose among alternative cost functions in a given application setting.

When a system contains more than a single adaptive element, analogous tools can be used by recognizing that the cost function of one adaptive element is not necessarily independent of the parameters of another element. The "cross sensitivity" is defined as the derivative of the cost function of one adaptive element to the parameters of another. This may be useful in terms of ordering a series of adaptive elements. Again speaking loosely, when two elements have low cross sensitivity, each operates independently of the other. When the cross sensitivities are large, there is significant interaction between the elements.

What is an Adaptive Element?

Definition: An *adaptive element* $\mathcal{E}$ is a (possibly nonlinear, generally time-varying) system with (1) input x_k, (2) ouput y_k, (3) parameter (vector) $\hat{\theta}$, (4) system function $f(\cdot)$ that defines the output in terms of the inputs $x_k, x_{k-1}, \ldots$ and the parameters $\hat{\theta}$ (and possibly also the previous outputs y_{k-j} for $j \geq 1$), and (5) a differentiable cost function $J(\hat{\theta})$. Because of our focus on communication receivers, we immediately specialize to the case where the cost function will be optimized via a gradient approach. Thus the parameter vector (denoted $\hat{\theta}_k$ at time k) will be adapted over time by

$$\hat{\theta}[k+1] = \hat{\theta}[k] - \mu \frac{\partial J(\hat{\theta}[k])}{\partial \hat{\theta}[k]} \tag{1}$$

where μ is a positive stepsize that helps to define the size of the update. Thus, the adaptive element is also characterized by (6) a stepsize (vector) μ. While (6) is not a necessary part of the model, its inclusion makes the statements of the results easier.

Example 1 (Phase Locked Loop) *A PLL is one of the most common adaptive receiver elements. In the noise free case, the input signal* $x(t) = \cos(2\pi f_0 t + \phi)$ *is presumed to be a sinusoid of known frequency* f_0 *and unknown phase* ϕ*. The output* $y(t)$ *is an estimate of* ϕ*, and is thus equal to the parameter vector* $\hat{\theta}$*. The cost function to be maximized is*

$$J_{PLL}(\hat{\theta}) = LPF\{x(t)\cos(2\pi f_0 t + \hat{\theta})\}.$$

where $LPF\{\cdot\}$ *represents an ideal low pass filter that removes all energy above* $3f_0/2$*. Straightforward trigonometric manipulation shows that this can be rewritten*

$$J_{PLL}(\hat{\theta}) = \frac{1}{2}\cos(\phi - \hat{\theta}). \tag{2}$$

Accordingly, the corresponding iterative algorithm is

$$\theta[k+1] = \theta[k] - \mu LPF\{x(kT)\sin(2\pi f_0 kT + \theta[k])\}, \tag{3}$$

which is a standard formula for the PLL [3].

Example 2 (Costas Loop) *With the same input, output, and system function as for the PLL, the cost for the Costas loop is*

$$J_{COSTAS}(\hat{\theta}) = \frac{1}{2}\cos^2(\phi - \theta). \tag{4}$$

The derivative of (4) is $\cos(\phi - \theta)\ \sin(\phi - \theta)$*. As above,* $\cos(\phi - \theta) = LPF\{2x(t)\cos(2\pi f_0 t + \theta)\}$*. Similarly,* $\sin(\phi - \theta) = -LPF\{2x(t)\sin(2\pi f_0 t + \theta)\}$*. Accordingly, the corresponding iterative algorithm is*

$$\begin{aligned}\theta[k+1] = \theta[k] - \mu LPF\{2x(kT)\sin(2\pi f_0 kT + \theta[k])\}\\ LPF\{2x(kT)\cos(2\pi f_0 kT + \theta[k])\}.\end{aligned}$$

Example 3 (Decision Directed PLL) *Let* $s(t)$ *be a pulse shaped signal created from a message where the symbols are chosen from some finite alphabet. At the transmitter,* $s(t)$ *is modulated by a carrier at frequency* f_0 *with unknown phase* ϕ*, and at the receiver it is demodulated by a sinusoid and then low pass filtered to create the input to the adaptive element*

$$x(t) = 2LPF\{s(t)\cos(2\pi f_0 t + \phi)\cos(2\pi f_0 t + \theta)\}. \tag{5}$$

When the phases ϕ *and* θ *are equal, then* $x(t) = s(t)$*. In particular,* $x(kT) = s(kT)$ *at the sample instants* $t = kT$*, where the* $s(kT)$ *are elements of the alphabet. On the other hand, if* $\phi \neq \theta$*, then* $x(kT)$ *will not be a member of the alphabet. The difference between what* $x(kT)$ *is, and what it should be, can be used to form a cost function and hence a phase tracking algorithm. The memoryless nonlinearity* $Q(z)$ *maps any real number* z *to the closest element of the symbol alphabet. The cost function for the decision directed method is*

$$J_{DD}(\theta) = \left(Q(x(kT)) - x(kT)\right)^2. \tag{6}$$

$\frac{dx(kT)}{d\theta}$ *can be calculated directly from (5) and the algorithm is:*

$$\begin{aligned}\theta[k+1] = \theta[k] - \mu\left(Q(x(kT)) - x(kT)\right)\\ LPF\{x(kT)\sin(2\pi f_0 kT + \theta[k])\}\end{aligned} \tag{7}$$

Example 4 (Minimization of the Cluster Variance) *The problem of clock recovery is to choose the sampling instants* τ*. The input to the element is the* M *times oversampled signal* $x(t)$*. If the combination of the pulse shape and the matched filter is Nyquist, then the value of the waveform is equal to the value of the data at the correct sampling times. If there is a training sequence, then this can be used as the basis for the cost. When there is no training, it is possible to substitute the cluster variance* $E\{(Q(x[k]) - x[k])^2\}$ *where* $x[k] = x(\frac{kT}{M} + \tau)$*. Since it is impossible to directly optimize an expectation, the goal is to find* τ *to minimize*

$$J_{CV}(\tau) = (Q(x[k]) - x[k])^2. \tag{8}$$

Applying the standard iterative solution yields

$$\tau[k+1] \;=\; \tau[k] + 2\mu(Q(x[k]) - x[k])\frac{dx[k]}{d\tau}.$$

To turn this into an algorithm that can be easily simulated, the derivative can be approximated numerically, one possibility is to use

$$\frac{dx[k]}{d\tau} \approx \frac{1}{2\delta}\left(x(\frac{kT}{M} + \tau[k] + \delta) - x(\frac{kT}{M} + \tau[k] - \delta)\right).$$

Example 5 (Output Energy Maximization) *With the same set up as in Example 4, another useful cost is defined by* $E\{x^2[k]\} = E\{x^2(\frac{kT}{M} + \tau)\}$. *This can be approximated using the instantaneous cost*

$$J_{OE}(\tau) = x^2[k], \tag{9}$$

which leads directly to the gradient algorithm

$$\tau[k+1] = \tau[k] + 2\mu x[k] \frac{dx[k]}{d\tau} \tag{10}$$

where the derivative can be approximated numerically.

The Sensitivity Function

The shape of the cost function near the optimal value θ^* can be used to gain insight into the convergence and tracking abilityies of an adaptive element. Consider the sensitivity function

$$\mathcal{S} = \frac{\partial J(\hat{\theta})}{\partial \hat{\theta}}. \tag{11}$$

Since the adaptive element is updated proportionally to $\mu\mathcal{S}$, $\mathcal{S}$ is directly proportional to the speed at which the adaptive parameter can change.

The cost function and the related sensitivity function $\mathcal{S}$ may be useful when comparing two possible adaptive elements that are both designed for the same task. For instance, suppose that the first element $\mathcal{E}_A$ is characterized by

$$x_k, y_k, \hat{\theta}_A, f(\cdot), J_A(\hat{\theta}_A), \mu_A$$

while the second adaptive element $\mathcal{E}_B$ is characterized by

$$x_k, y_k, \hat{\theta}_B, f(\cdot), J_B(\hat{\theta}_B), \mu_B.$$

Since the inputs, outputs, and system functions are the same for both elements, both are designed for the same problem setting. Local properties of the cost and sensitivity functions may be useful in specifying desirable properties of the two elements and may provide a way to talk about the performance of $\mathcal{E}_A$ and $\mathcal{E}_B$ in a concrete way. Since the cost functions J_A and J_B are different, they may have different stationary points θ_A^* and θ_B^*. If one of these stationary points leads to better system performance then it may be preferable to the other. If the region in which the cost function is unimodal is larger for one than the other, then it may be preferred because initialization of the parameter $\hat{\theta}$ will be less critical. If the magnitude of the sensitivity function $|\mathcal{S}_A|$ is greater than $|\mathcal{S}_B|$, then the element $\mathcal{E}_A$ may be preferable because it will likely have faster convergence.

Example 6 *The error surfaces for the three carrier recovery methods in Examples 1, 2, and 3 are shown in Figure 1. Observe that the fixed points and the regions of convergence differ for the three methods, providing valuable information about the behavior of the algorithms. When the error surface is flat, the sensitivity is zero, and the algorithm is at a fixed point.*

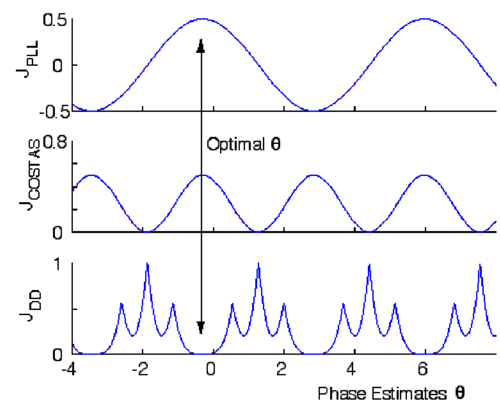

Figure 1: The error surfaces for the PLL (2) in the top plot, the Costas loop (4) in the middle plot, and the Decision Directed method (6) in the bottom plot.

The Cross-Sensitivity Function

Many receivers are designed with multiple cascaded adaptive elements. The cross-sensitivity function quantifies how the cost function of one element depends on the parameter of another, and gives a measure of the degree of interaction between the different elements.

To be concrete, consider two adaptive elements $\mathcal{E}_A$ and $\mathcal{E}_B$ which are characterized by

$$\begin{array}{l} \mathcal{E}_A : x_k^A, y_k^A, \hat{\theta}_A, f_A(\cdot), J_A(\hat{\theta}_A), \mu_A \\ \mathcal{E}_B : x_k^B, y_k^B, \hat{\theta}_B, f_B(\cdot), J_B(\hat{\theta}_B), \mu_B \end{array}$$

Then the *cross sensitivity* defines how a change in element A effects the cost function of element B

$$\mathcal{S}_{AB} = \frac{\partial J_B(\hat{\theta}_B)}{\partial \hat{\theta}_A}. \tag{12}$$

The cross sensitivity of $\mathcal{E}_B$ to $\mathcal{E}_A$ is defined analogously. Sometimes (as in Example 8), it is possible to merge the two adaptive elements into a single (vector) element. The behavior of the coupled system can be understood by looking at the multidimensional error surface. Sometimes, the form of the individual elements precludes expressing the pair as both operating on a single cost function. In this case, an extended notion of error surface considers two families of surfaces $J_A(\hat{\theta_A})|_{\hat{\theta}_B}$ (plotted over a range of values $\hat{\theta}_B$) and the analogous $J_B(\hat{\theta_B})|_{\hat{\theta}_A}$.

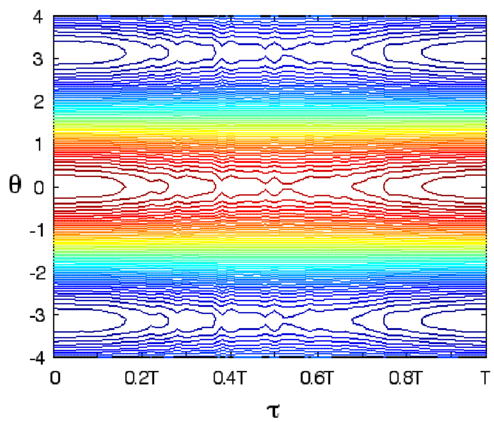

Figure 2: Plot of $\mathcal{S}_{BA}$ as a function of θ and τ. A SRRC pulse shape is assumed with a rolloff factor of 0.5. There is a single minimum within each 2π.

Example 7 *Consider a receiver that uses the PLL of Example 1 (system $\mathcal{E}_A$ with $\hat{\theta}_A = \theta$ of (3)) in cascade with the OE method of clock recovery (system $\mathcal{E}_B$ with $\hat{\theta}_B = \tau$ of (10)). Since $\mathcal{S}_{BA} = 0$, the adaptive element $\mathcal{E}_B$ has no impact on $\mathcal{E}_A$; the carrier recovery operates independently of the clock recovery. On the other hand,*

$$\mathcal{S}_{AB} = \frac{\partial x^2(\frac{kT}{M} + \tau[k])}{\partial \theta[k]}$$

where

$$x(t) = LPF\{s(t)\cos(2\pi f t + \phi)\cos(2\pi t + \theta)\}.$$

This simplifies to

$$\mathcal{S}_{AB} = x(kT/M + \tau)s(kT/M + \tau)\cos(\phi - \theta).$$

When the decisions are correct $s(kT/M+\tau) = x(kT/M+\tau)$, and this is just a constant times $\mathcal{S}_B$. Except for those θ where $\cos(\phi - \theta) = 0$, the minimizing points of $\mathcal{E}_B$ are unchanged by the presence of $\mathcal{E}_A$. This is plotted in Figure 2.

Example 8 *Suppose that $\mathcal{E}_A$ is the DD PLL of Example 3 and that $\mathcal{E}_B$ is the CV method of clock recovery in Example 4. Because the form of J_{DD} in (6) is the same as that of J_{CV} in (8), $\mathcal{S}_{AB} = \mathcal{S}_B$ and $\mathcal{S}_{BA} = \mathcal{S}_A$. This is plotted in Figure 3.*

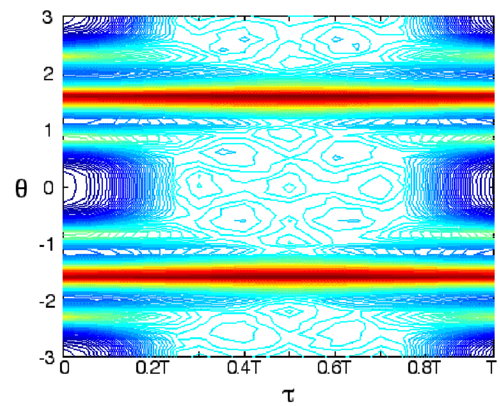

Figure 3: Plot of $\mathcal{S}_{BA}$ as a function of θ and τ. A SRRC pulse shape is assumed with a rolloff factor of 0.5. There are many minima, and the behavior of the algorithm may be unreliable.

Thus, when the cross sensitivity functions are nontrivial, the two elements may interfere. The cross sensitivity between adaptive elements gives a way to qualitatively and quantitatively talk about the interaction between adaptive elements that operate simultaneously within a given system. The cross sensitivity can be used to study and understand possible re-arrangements of the system.

References

[1] R. D. Gitlin, J. F. Hayes, and S. B. Weinstein, *Data Communication Principles*, New York, Plenum Press, 1992.

[2] C. R. Johnson and W. A. Sethares, *Telecommunciation Breakdown*, Prentice-Hall (2003).

[3] J.G. Proakis, *Digital Communications*, New York, McGraw-Hill, 1995.