PAST INPUT RECONSTRUCTION IN BACKWARD CONSISTENT FAST LEAST-SQUARES ALGORITHMS

Phillip A. Regalia
Département Signal & Image
Institut National des Télécommunications
9, rue Charles Fourier
F-91011 Evry cedex France
e-mail: regalia@galaxie.int-evry.fr

ABSTRACT

We present an analytic solution to the past input reconstruction problem, which consists in describing all past input sequences which would give rise to a given set of variables in fast least-squares algorithms, whenever the variables in question are reachable.

1 INTRODUCTION

Let $\mathbf{u}_k$ be a sequence of row vectors, each with $M+1$ elements, and stack these one atop another to build a data matrix $\mathbf{U}(n)$, with $\mathbf{u}_n$ as the top row. With $\Lambda(n) = \mathrm{diag}[1, \lambda, \dots, \lambda^n]$, recursive least-squares filtering algorithms often invole the time-propagation of the covariance matrix $\mathbf{P}(n) = \mathbf{U}^t(n)\,\Lambda(n)\,\mathbf{U}(n)$, or its inverse, or its Cholesky factor, etc. Fast least-squares algorithms may be developed when the vectors $\mathbf{u}_n$ derive from a delay line, and the resulting algorithms feature order M complexity in both storage and computation. The matrix recursions involving $\mathbf{P}(n)$ are replaced by a prediction section, which takes the form of a time-recursive computation

$$\xi(n) = T[\xi(n-1), u_n]$$

in which u_n is a scalar input sample, $\xi(\cdot)$ is the state vector which collects all variables that need be written for storage, and $T[\cdot,\cdot]$ is a nonlinear map which implements the fast least-squares prediction subroutine at each time iteration.

Suppose the past input u_n, u_{n-1}, u_{n-2} ..., is allowed to vary arbitrarily, and let $\mathcal{S}_i$ be the set of state variables $\xi(n)$ that are reachable in exact arithmetic. It is known [1]–[3] that unstable error propagation is possible only if the finite precision version of $\xi(\cdot)$ exits $\mathcal{S}_i$, so that $\mathcal{S}_i$ furnishes a stability domain. Deducing necessary conditions for a candidate state $\xi(\cdot)$ to belong to $\mathcal{S}_i$ involved exploiting known least-squares consistency conditions [2], [3]; showing these conditions to be sufficient involved further labor [4]. But by definition of $\mathcal{S}_i$, if a given state $\xi(\cdot)$ is indeed reachable, then it must be possible to place in evidence some past input sequence u_n, u_{n-1}, u_{n-2} ..., which gives rise to this state. We solve here the past input reconstruction problem, which consists in describing all valid past inputs for a given state, whenever the state is reachable. This complements the stability domain concept initiated in [1].

2 PROBLEM STRUCTURE

In fast least-squares algorithms the input vector derives from a scalar sequence passed through a delay line:

$$\mathbf{u}_n = [u_n \;\; u_{n-1} \;\; \cdots \;\; u_{n-M}].$$

If $u_n = 0$ for $n < 0$, the matrix $\mathbf{U}(n)$ then assumes a "prewindowed" Hankel structure:

$$\mathbf{U}(n) = \begin{bmatrix} \mathbf{u}_n \\ \mathbf{u}_{n-1} \\ \vdots \\ \mathbf{u}_M \\ \vdots \\ \mathbf{u}_1 \\ \mathbf{u}_0 \end{bmatrix} = \begin{bmatrix} u_n & u_{n-1} & \cdots & u_{n-M} \\ u_{n-1} & u_{n-2} & \cdots & u_{n-M-1} \\ \vdots & \iddots & \iddots & \vdots \\ u_M & \cdots & u_1 & u_0 \\ \vdots & \iddots & \iddots & 0 \\ u_1 & u_0 & \iddots & \vdots \\ u_0 & 0 & \cdots & 0 \end{bmatrix} \qquad (1)$$

Let us introduce the correlation lags

$$r_k = \sum_{i=0}^{n} u_{n-i}\, u_{n-i-k}, \qquad k = 0, 1, \dots, M, \qquad (2)$$

and likewise rename the most recent input samples as

$$x_1 = u_n, \quad x_2 = u_{n-1}, \quad \dots \quad x_M = u_{n-M+1}. \qquad (3)$$

Then for any n, one may check that the gramian of $\mathbf{U}(n)$, using $\lambda = 1$, takes the form

$$\mathbf{P}(n) = \mathbf{U}^t(n)\,\mathbf{U}(n)$$
$$= \begin{bmatrix} r_0 & r_1 & \cdots & r_M \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_M & \cdots & r_1 & r_0 \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \iddots & 0 & x_1 \\ 0 & \iddots & \iddots & \vdots \\ 0 & x_1 & \cdots & x_M \end{bmatrix}^2 \qquad (4)$$

The matrix $\mathbf{P}(n)$ is completely specified by $2M+1$ values, namely $r_0, \dots, r_M$ and $x_1 \dots, x_M$.

When using a forgetting factor λ, with $\lambda < 1$, the matrix $\mathbf{P}(n)$ becomes

$$\mathbf{P}(n) = \mathbf{U}^t(n)\,\Lambda(n)\,\mathbf{U}(n) \qquad (5)$$

Set $\mathbf{L} = \mathrm{diag}[1, \lambda^{1/2}, \dots, \lambda^{M/2}]$; since $\mathbf{U}(n)$ is a Hankel matrix,

$$\Lambda^{1/2}(n)\,\mathbf{U}(n) = \overline{\mathbf{U}}(n)\,\mathbf{L}^{-1}$$

in which $\overline{\mathbf{U}}(n)$ is a Hankel matrix akin to (1), but built from the sequence

$$\bar{u}_{n-k} = \lambda^{k/2} u_{n-k}. \tag{6}$$

As such, the matrix $\mathbf{P}(n)$ from (5), once multiplied from the left and right by the matrix $\mathbf{L}$, will assume the same structure as if $\lambda = 1$ had been used [cf. (4)], and the past input had been exponentially weighted, as in (6). As this removes the influence of λ, we may set $\lambda = 1$ with no loss of generality.

We now review more common parametrizations of $\mathbf{P}(n)$.

2.1 Fast Transversal Filters

The fast transversal equations (with their many variants) are well defined only when $\mathbf{P}(n)$ is invertible. The inverse $\mathbf{P}^{-1}$ (time index n suppressed) has low displacement rank according to [2]

$$\begin{bmatrix} \mathbf{P}^{-1} & \mathbf{0} \\ \mathbf{0}^t & 0 \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{0}^t \\ \mathbf{0} & \mathbf{P}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_M \\ 0 \end{bmatrix} [\mathbf{A}_M^t \;\; 0] + \begin{bmatrix} \mathbf{C}_M \\ 0 \end{bmatrix} [\mathbf{C}_M^t \;\; 0] - \begin{bmatrix} 0 \\ \mathbf{B}_M \end{bmatrix} [0 \;\; \mathbf{B}_M^t] \tag{7}$$

in which the vectors $\mathbf{A}_M$, $\mathbf{B}_M$, and $\mathbf{C}_M$ contain, respectively, normalized versions of the forward prediction error filter, the backward prediction error filter, and the Kalman gain vector. We refer to Slock [2] for more detail. These algorithms perform time updates not on the matrix $\mathbf{P}^{-1}(n)$, but on the corresponding generator vectors $\mathbf{A}_M(n)$, $\mathbf{B}_M(n)$, and $\mathbf{C}_M(n)$; these variables in turn yield the state vector $\xi(n)$.

2.2 Order Recursive Algorithms

Suppose $\mathbf{P}$ is truncated to its $(k+1)\times(k+1)$ principal submatrix; the resulting matrix, once inverted and displaced akin to (7), yields generator vectors $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$, each of $k+1$ elements. For any order k, set

$$\begin{aligned} A_k(z) &= [1 \; z \; \cdots \; z^k]\,\mathbf{A}_k \\ B_k(z) &= [1 \; z \; \cdots \; z^k]\,\mathbf{B}_k \\ C_k(z) &= [1 \; z \; \cdots \; z^k]\,\mathbf{C}_k \end{aligned}$$

These polynomials (evaluated at a common time index n) are known to be related by the order recursion [5]

$$\begin{bmatrix} A_{k+1}(z) \\ C_{k+1}(z) \\ B_{k+1}(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{\cos\phi_k} & & \frac{\sin\phi_k}{\cos\phi_k} \\ & 1 & \\ \frac{\sin\phi_k}{\cos\phi_k} & & \frac{1}{\cos\phi_k} \end{bmatrix} \begin{bmatrix} 1 & & \\ & \frac{\sin\theta_k}{\cos\theta_k} & \frac{1}{\cos\theta_k} \\ & \frac{1}{\cos\theta_k} & \frac{\sin\theta_k}{\cos\theta_k} \end{bmatrix} \times \begin{bmatrix} A_k(z) \\ C_k(z) \\ z\,B_k(z) \end{bmatrix} \tag{8}$$

in which $\sin\phi_k$ is the correlation coefficient between the normalized forward and backward prediction errors of degree k; and $\sin\theta_k$ is the angle normalized backward prediction error of degree k, divided by the square-root of the corresponding backward prediction error energy. These rotation angles appear in fast QR algorithm studied in [3], yielding the state vector $\xi(n)$, and many other variants may be found in fast QR/lattice algorithms [5]–[7].

2.3 Shift Invariance

Suppose the parameter values $\{r_k\}_{k=0}^M$ and $\{x_k\}_{k=1}^M$ are reachable at time n, i.e., there exists some input sequence $\{u_i\}_{i=0}^n$ fulfilling (2) and (3). Then these same values are reachable at time $n+1$, by applying a causal shift to the input sequence. Conversely, any parameter set $\{r_k\}_{k=0}^M$ and $\{x_k\}_{k=1}^M$ reachable at time $n+1$ is also reachable at time n, provided the starting time is pushed back to $i=-1$. The set of *asymptotically reachable* parameters may be understood as those reachable by fixing the starting time at $i=0$ and letting the final time extend to $n=+\infty$, or equivalently, by fixing the final time to $n=-1$ and letting the starting time extend back to $i=-\infty$.

Upon adopting the latter convention, the z-transform of any valid past input sequence takes the form

$$U(z) = \sum_{i=1}^{\infty} u_{-i} z^i, \qquad |z| < 1, \tag{9}$$

which yields a function analytic in $|z|<1$. Moreover, the set of parameters $\{r_k\}$ and $\{x_k\}$ reachable at time $n=-1$ corresponds precisely to the set of valid initial conditions for the fast least-squares algorithm to proceed correctly from time $n=0$ onward. The past input reconstruction problem is then:

Problem 1 *Given the structured matrix*

$$\mathbf{P}(-1) = \begin{bmatrix} r_0 & r_1 & \cdots & r_M \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_M & \cdots & r_1 & r_0 \end{bmatrix} - \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \iddots & 0 & x_1 \\ 0 & \iddots & \iddots & \vdots \\ 0 & x_1 & \cdots & x_M \end{bmatrix}^2$$

find all anti-causal functions as in (9) which satisfy the interpolation conditions

$$\begin{aligned} u_{-k} &= x_k, & k &= 1,2,\ldots,M; & (10) \\ \sum_{i=1}^{\infty} u_{-i}\,u_{-i-k} &= r_k, & k &= 0,1,\ldots,M. & (11) \end{aligned}$$

This problem first arose in model reduction in Mullis and Roberts [8]; see also [9] and [10]. These works claim that a solution exists if and only if $\mathbf{P}(-1)$ is nonnegative definite. Connections to classical interpolation theory surfaced in [11] and [12], from which one may show that a solution need not exist when $\mathbf{P}(-1)$ is positive semi-definite.

3 A RELATED INTERPOLATION PROBLEM

Let $\mathcal{Z}$ be the shift matrix with ones on the subdiagonal and zeros elsewhere. The matrix $\mathbf{P}(-1)$ has low displacement rank, and its displacement residue $\mathbf{P}(-1) - \mathcal{Z}\mathbf{P}(-1)\mathcal{Z}^t$ becomes

$$\mathbf{P}(-1) - \mathcal{Z}\mathbf{P}(-1)\mathcal{Z}^t = \begin{bmatrix} r_0 & r_1 & \cdots & r_M \\ r_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r_M & 0 & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ x_1 \\ \vdots \\ x_M \end{bmatrix} [\cdot]^t$$

$$= \begin{bmatrix} \sqrt{r_0} \\ r_1/\sqrt{r_0} \\ \vdots \\ r_M/\sqrt{r_0} \end{bmatrix} [\cdot]^t - \begin{bmatrix} 0 \\ r_1/\sqrt{r_0} \\ \vdots \\ r_M/\sqrt{r_0} \end{bmatrix} [\cdot]^t - \begin{bmatrix} 0 \\ x_1 \\ \vdots \\ x_M \end{bmatrix} [\cdot]^t \tag{12}$$

where "[·]" means "repeat the previous vector". Now, behind most any displacement structure lurks an interpolation problem [13], [14]; that corresponding to (12) may be introduced as follows.

Let $\mathbf{S}(z)$ be a 2×1 vector-valued Schur function, meaning that $\mathbf{S}(z)$ is analytic in $|z|<1$ and contractive, i.e., $\|\mathbf{S}(z)\|<1$ in $|z|<1$, where $\|\cdot\|$ denotes the Euclidean norm. Let us set

$$a(z) \triangleq \sqrt{r_0} + \frac{r_1}{\sqrt{r_0}} z + \cdots + \frac{r_M}{\sqrt{r_0}} z^M, \qquad (13)$$

as well as $\left[\begin{smallmatrix} c(z) \\ b(z) \end{smallmatrix}\right] = \mathbf{S}(z)\, a(z)$. We then have:

Problem 2 *Given the parameters* $\{r_k\}_{k=0}^{M}$ *and* $\{x_k\}_{k=1}^{M}$*, find a Schur function* $\mathbf{S}(z)$ *such that the resulting* $b(z)$ *and* $c(z)$ *assume the forms*

$$\begin{aligned} c(z) &= 0 + x_1 z + x_2 z^2 + \cdots + x_M z^M + O_1(z^{M+1}) && (14) \\ b(z) &= 0 + \frac{r_1}{\sqrt{r_0}} z + \cdots + \frac{r_M}{\sqrt{r_0}} z^M + O_2(z^{M+1}) && (15) \end{aligned}$$

where $O(z^{M+1})$ *denotes a function analytic in* $|z|<1$ *which vanishes* $M+1$ *times at* $z=0$.

This problem admits a solution $\mathbf{S}(z)$ if and only if a certain Pick matrix is nonnegative definite [15]; that corresponding to the present problem is simply $\mathbf{P}(-1)$ from (12).

Since $b(z)$ and $c(z)$ both vanish at $z=0$, while $a(z)$ does not, we see that any solution $\mathbf{S}(z)$ to Problem 2 must vanish at $z=0$. This allows us to write $\mathbf{S}(z) = \left[\begin{smallmatrix} zS_1(z) \\ zS_2(z) \end{smallmatrix}\right]$. It is known [15] that whenever solutions exist, then lossless solutions exist, where lossless refers to a Schur function which has unit norm along the unit circle $z=e^{j\omega}$:

$$|S_1(e^{j\omega})|^2 + |S_2(e^{j\omega})|^2 = 1, \qquad \text{for all } \omega. \qquad (16)$$

Proposition 3 *Let* $\mathbf{S}(z)$ *be a lossless solution to Problem 2. If the resulting* $zS_2(z)$ *obeys the constraint*

$$1 - zS_2(z) \neq 0, \qquad \textit{for all } |z| = 1, \qquad (17)$$

then the function

$$U(z) = \sqrt{r_0}\, \frac{zS_1(z)}{1 - zS_2(z)} \qquad (18)$$

is a solution to Problem 1. Moreover, all solutions to Problem 1 may be generated in this way.

For a proof, see [11]. In case (17) is violated, i.e., $e^{j\omega_0} S_2(e^{j\omega_0}) = 1$ for some value ω_0, then (16) gives $S_1(e^{j\omega_0}) = 0$, producing a pole-zero cancellation on the unit circle in $U(z)$. This possibility did not appear in [8]–[10], which explains the shortcoming of their claimed sufficient conditions.

4 CONSTRUCTING $\mathbf{S}(z)$

Solutions to Problem 2 may be constructed by using a Schur algorithm; that to follow is adapted from [16].

We begin with the data array

$$\mathbf{G} = \begin{bmatrix} \sqrt{r_0} & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \\ 0 & x_1 & x_2 & \cdots & x_M \\ 0 & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \end{bmatrix}, \qquad (19)$$

which contains the leading terms of the functions $a(z)$, $c(z)$, and $b(z)$ from (13), (15), and (14).

1. Shift the first row of the array (19) one position to the right:
$$(19) \xrightarrow{z} \begin{bmatrix} 0 & \sqrt{r_0} & r_1/\sqrt{r_0} & \cdots & r_{M-1}/\sqrt{r_0} \\ 0 & x_1 & x_2 & \cdots & x_M \\ 0 & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \end{bmatrix}.$$
2. Choose a hyperbolic rotation to knock off the second element of the first nonzero column. In the first pass, this appears as
$$\begin{aligned} &\begin{bmatrix} 1/\cos\theta_0 & \sin\theta_0/\cos\theta_0 & \\ \sin\theta_0/\cos\theta_0 & 1/\cos\theta_0 & \\ & & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 0 & \sqrt{r_0} & r_1/\sqrt{r_0} & \cdots & r_{M-1}/\sqrt{r_0} \\ 0 & x_1 & x_2 & \cdots & x_M \\ 0 & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \end{bmatrix} \\ &= \begin{bmatrix} 0 & y_1 & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ 0 & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \end{bmatrix}, \end{aligned}$$
in which $y_1 = \sqrt{r_0 - x_1^2}$ and $\sin\theta_0 = -x_1/\sqrt{r_0}$.
3. Choose a hyperbolic rotation to knock off the third element of the first nonzero column. In the first pass, this appears as
$$\begin{aligned} &\begin{bmatrix} 1/\cos\phi_0 & & \sin\phi_0/\cos\phi_0 \\ & 1 & \\ \sin\phi_0/\cos\phi_0 & & 1/\cos\phi_0 \end{bmatrix} \\ &\times \begin{bmatrix} 0 & y_1 & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ 0 & r_1/\sqrt{r_0} & r_2/\sqrt{r_0} & \cdots & r_M/\sqrt{r_0} \end{bmatrix} \\ &= \begin{bmatrix} 0 & y_2 & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \end{bmatrix}, \end{aligned} \qquad (20)$$
in which $y_2 = \sqrt{y_1^2 - (r_1^2/r_0)}$ and $\sin\phi_0 = -(r_1/\sqrt{r_0})/y_1$.
4. Replace the array (19) with (20) and reiterate the above M–1 times, to eliminate all the elements of the second and third rows.

This procedure continues M full iterations yielding $|\sin\theta_k| < 1$ and $|\sin\phi_k| < 1$ if and only if the matrix $\mathbf{P}(-1)$ is positive definite [16]. If $\mathbf{P}(-1)$ is positive semi-definite, of rank $k < M+1$, the procedure terminates after k stages, yielding $|\sin\theta_{k-1}| = 1$ or $|\sin\phi_{k-1}| = 1$ [16].

A flowgraph of this operation for the positive definite case, applied to the functions $a(z)$, $c(z)$, and $b(z)$, appears in Figure 1, for the case $M=3$. Each successive stage introduces another leading zero into the three functions. The flowgraph of

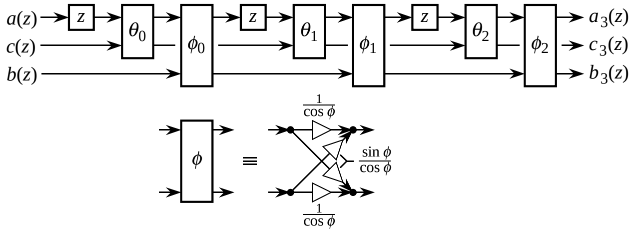

Figure 1: Illustrating the Schur algorithm, for $M = 3$.

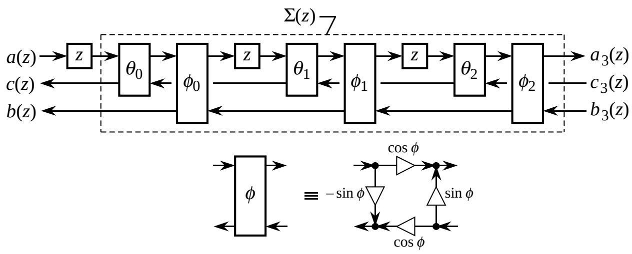

Figure 2: Redrawing of Figure 1, using planar rotations.

Figure 1 can be rearranged into that of Figure 2, by reversing the flow direction of the two lower branches; the relations between the various intermediate signals appearing in Figures 1 and 2 are preserved. Upon closing the right-hand port in a lossless load, according to

$$\begin{bmatrix} c_M(z) \\ b_M(z) \end{bmatrix} = z\,\mathbf{S}_L(z)\,a_M(z),$$

the resulting function mapping $a(z)$ to $\left[\begin{smallmatrix} c(z) \\ b(z) \end{smallmatrix}\right]$ is lossless, and all lossless solutions to Problem 2 are exhausted by varying $\mathbf{S}_L(z)$ over all lossless possibilities (e.g., [15]). A realization of $U(z)$ as per (18), finally, is obtained by closing the input port and scaling the remaining output, as in Figure 3.

We now relate the rotation angles in Figure 2 to the state variables of fast least-squares algorithms. The following identity may be attributed to Lev-Ari *et al*. [5]:

Identity 4 *The rotation angles* $\{\theta_k\}$ *and* $\{\phi_k\}$ *of the order recursion (8) are precisely the angles determined from the above Schur algorithm.*

These angles make an explicit appearance in, e.g., the fast QR algorithm studied in [3],[1] and can be inferred from other minimal lattice and QR algorithms (e.g., [5], [6], [7]).

This then specifies the fixed rotation angles building $\Sigma(z)$ in Figure 3. As for the lossless load $\mathbf{S}_L(z)$, the simplest choice is a constant: $\mathbf{S}_L = \left[\begin{smallmatrix} \sin\alpha \\ \cos\alpha \end{smallmatrix}\right]$, where α may be varied freely. If $\mathbf{P}(-1)$ is positive definite, then one may show that finitely many values of α exist for which the realization of Figure 3 may lose observability or controllability, and thus a continuum of values exists for which the realization is minimal (no pole-zero cancellation). Any such value of α must give a $zS_2(z)$ for which (17) is satisfied.

[1] The angles θ_k in Figure 2 are precisely those of [3], but the angles ϕ_k are denoted by ϕ_{k+1} in [3]. The index on ϕ is decremented in this paper so that rotations within a common section of Figure 2 take the same index.

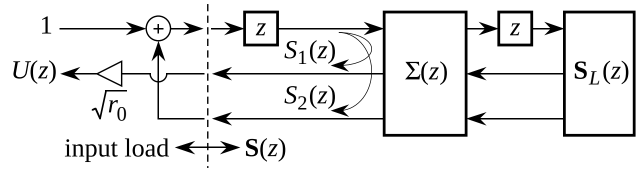

Figure 3: Closing the loop to realize the function $U(z)$.

References

[1] P. A. Regalia, “Numerical stability issues in fast least-squares adaptation algorithms,” *Opt. Engr.*, **31**, pp. 1144–1152, 1992.

[2] D. T. M. Slock, “The backward consistency concept and round-off error propagation dynamics in recursive least-squares algorithms,” *Opt. Engr.*, **31**, pp. 1153–1169, June 1992.

[3] P. A. Regalia, “Numerical stability properties of a QR-based fast least-squares algorithm,” *IEEE Trans. Signal Processing*, **41**, pp. 2096–2109, June 1993.

[4] P. A. Regalia and F. Desbouvries, “Displacement structures of covariance matrices, lossless systems, and numerical algorithm design,” *SIAM J. Matrix Analysis and Applications*, **16**, pp. 536–564, April 1995.

[5] H. Lev-Ari, T. Kailath, and J. Cioffi, “Least-squares adaptive lattice and transversal filters: A unified geometric theory,” *IEEE Trans. Info. Theory*, **30**, pp. 228–236, March 1984.

[6] D. T. M. Slock, “Reconciling fast RLS lattice and QR algorithms,” *Proc. ICASSP-90*, Albuquerque, NM; pp. 1591–1594.

[7] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, “Computationally efficient QR decomposition approach to least squares adaptive filtering,” *IEE Proc.-F*, **138**, pp. 341–353, 1991.

[8] C. T. Mullis and R. A. Roberts, “The use of second-order information in the approximation of discrete-time linear systems,” *IEEE Trans. Acous., Speech, Sig. Proc.*, **24**, pp. 226–238, 1976.

[9] Y. Inouye, “Approximation of multivariable linear systems with impulse response and autocorrelation sequences,” *Automatica*, **19**, pp. 265–277, 1983.

[10] A. M. King, U. B. Desai, and R. E. Skelton, “A generalized approach to q-Markov covariance equivalent realizations for discrete systems,” *Automatica*, **24**, pp. 507–515, 1988.

[11] P. A. Regalia, M. Mboup, and M. Ashari, “A class of first- and second-order interpolation problems in model reduction,” *Archiv für Elek. und Übertrag.*, **49**, pp. 332–343, Sept. 1995.

[12] D. Alpay, V. Bolotnikov, and Ph. Loubaton, “On tangential $\mathcal{H}_2$ interpolation with second-order constraints,” *Integral Equations and Operator Theory* (to appear).

[13] D. Alpay, P. Dewilde, and H. Dym, “On the existence and construction of solutions to the partial lossless inverse scattering problem,” *IEEE Trans. Inf. Th.*, **35**, pp. 1184–1205, Nov. 1989.

[14] T. Kailath and A. H. Sayed, “Displacement structure: Theory and applications,” *SIAM Review*, **37**, pp. 297–386, Sept. 1995.

[15] H. Dym, *J-Contractive Matrix Functions, Reproducing Kernel Hilbert Spaces, and Interpolation*, CBMS Regional Conf. Series, **71**, Amer. Math. Soc., Providence, RI, 1989.

[16] H. Lev-Ari and T. Kailath, “Lattice filter parametrization and modeling of nonstationary processes,” *IEEE Trans. Information Theory*, **30**, pp. 2–16, p. 878, 1984.