LU and spectral factorization

Patrick Dewilde

TUM Inst. for Advanced Study

IAS TUM Institute for Advanced Study

LU-factorization

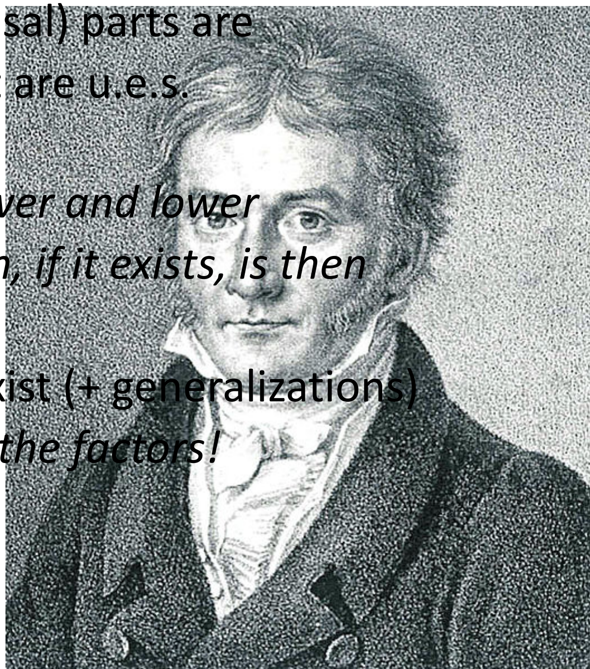
Starting point: a quasi-separable representation of a mixed transfer operator:

$$T = C_c(I - ZA_c)^{-1}ZB_c + D + C_a(I - Z^*A_a)^{-1}Z^*B_a.$$

Working assumption: both lower (causal) and upper (anti-causal) parts are "regular" – i.e. they have input and output normal forms that are u.e.s.

What is an LU-factorization? We require: *T=LU such that L is lower and lower invertible, and U is upper and upper invertible. The factorization, if it exists, is then automatically minimal.*

Our goal will be to find NS Conditions for the factorization to exist (+ generalizations) *and to derive a (new) numerically stable algorithm to compute the factors!*



Gauss, 1777-1855

IAS TUM Institute for Advanced Study

Motivation

1. Generalizes the finite matrix case.
2. Excludes “mixed” factorizations, s.a. *$T=UT_oV$ – of the “URV” type.* These are non-minimal factorizations, which allow the computation of Moore-Penrose inverses (see D-vdV textbook)
3. Makes the factorization (if it exists) essentially unique, forces, in the LTI case, the matching of internal zeros/poles and external zeros/poles:

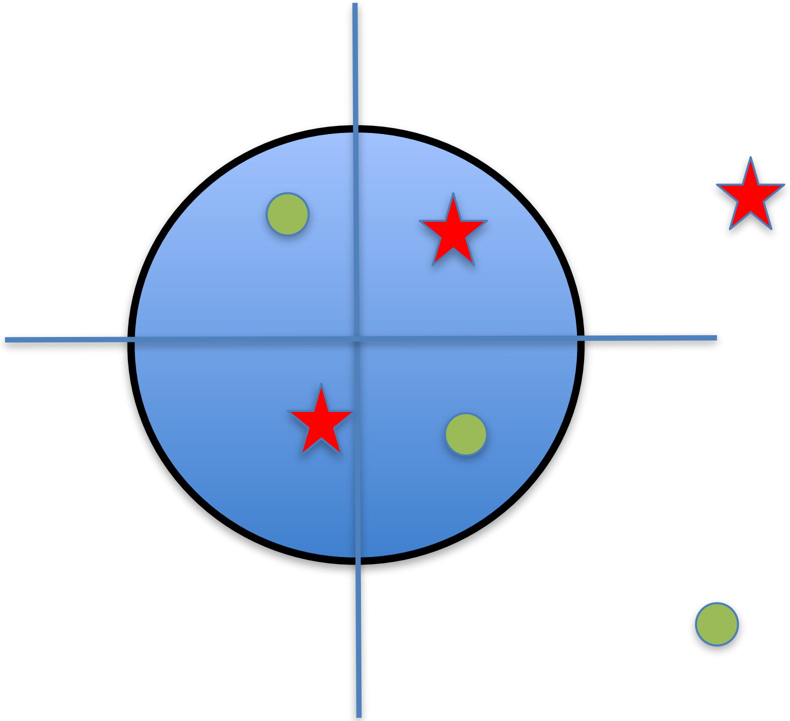
$$
\begin{aligned}
\mathrm{To}[\cdots 0, -\tfrac{1}{2}, \boxed{\tfrac{7}{6}}, -\tfrac{1}{3}, 0 \cdots] &= \mathrm{To}[\cdots 0, -\tfrac{1}{2}, \boxed{1}, 0 \cdots] \cdot \mathrm{To}[\cdots 0, \boxed{1}, -\tfrac{1}{3}, 0 \cdots] \\
&= \mathrm{To}[\cdots 0, -\tfrac{1}{2}, \boxed{\tfrac{1}{6}}, 0 \cdots] \cdot \mathrm{To}[\cdots 0, \boxed{1}, -2, 0 \cdots].
\end{aligned}
$$

IAS TUM Institute for Advanced Study

The naive (LTI) way

Match poles and zeros inside/outside:

LU-factorization = spectral factorization
For an LU-factorization to exist in the Toeplitz-case, inside and outside degrees in zeros and poles must match!



Can spectral factorization be generalized to the matrix case?

The algorithm

$$T = C_c(I - ZA_c)^{-1}ZB_c + D + C_a(I - Z^*A_a)^{-1}Z^*B_a.$$

(minimal "regular" realizations of lower and upper parts)

Step 1: put (or assume) $[A_a \;\; B_a]$ isometric (assumption: "regularity"!). Let W be a unitary operator defined by the completed unitary realization

$$W \sim_c \begin{bmatrix} A_a^* & B_W \\ B_a^* & D_W \end{bmatrix}.$$

and let $T_u = TW$. Then T_u is causal (lower) and realized by

$$\begin{bmatrix} A_u & B_u \\ C_u & D_u \end{bmatrix} := \left[\begin{array}{cc|c} A_c & B_cB_a^* & B_cD_W \\ 0 & A_a^* & B_W \\ C_c & DB_a^* + C_aA_a^* & DD_W + C_aB_W \end{array}\right].$$

(this step is sometimes automatic, but may necessitate an extra square-root algorithm}

IAS TUM Institute for Advanced Study

Step 2: outer-inner factorization of T_u

original · Q^* factor = R factor

$$\begin{bmatrix} A_u Y & B_u \\ C_u Y & D_u \end{bmatrix} \cdot \begin{bmatrix} C_n^* & A_V^* & C_V^* \\ D_n^* & B_V^* & D_V^* \end{bmatrix} = \begin{bmatrix} 0 & Y^{<-1>} & B_o \\ 0 & 0 & D_o \end{bmatrix}$$

recursive factor ($C_u Y$, D_u)

kernel! (D_n^*)

result: V (B_V^*, D_V^*)

next rec. factor (0)

result: outer factor T_o (D_o)

$$T_u = T_0 V$$

For a solution to exist:

- kernel should be zero!
- recursive factor should be boundedly invertible!
- A_V should be u.e.s.!

but the (linear recursive) RQ-algorithm will always go through, allowing the checking

(in the finite matrix case, only the first one is relevant)

Step 3: determine the result

All the data needed for the result is now available! The structure of the outer-inner factorization forces *Y* to be of the form (see further!):

$$Y = \begin{bmatrix} Y_1 \\ R \end{bmatrix}$$

with *R* (block diagonal) square and bounded invertible. Then we have:

$$U = I + F(I - Z^* A_a)^{-1} Z^* B_a$$

with

$$F = -\left(C_W R A_V^* + D_W B_V^*\right)\left(R^{<-1>}\right)^{-1}.$$

and also

$$U^{-1} = I - F\left(I - Z^*(A_a - B_a F)\right)^{-1} Z^* B_a = I - F R^{<-1>}(I - Z^* A_V^*)^{-1} Z^* R^{-1} B_a.$$

showing that if all conditions are satisfied, *U* is bounded and bounded invertible

(W determines the "poles", V determines the "zeros"!)

how does the magic work?

This is the strategy:

1. push T to causality by using the reachability dynamics (A_a,B_a) of the anticausal part – in input normal form Inner W). This will leaves the dynamics of the inverse system unchanged; *this defines a lower system* T_u, *which has a state transition operator combining the causal and anticausal parts of the original system.*
2. determine the anticausal dynamics of the inverse of the resulting T_u – this is obtained through an outer-inner factorization of $T_u=T_oV$.
3. now we have to establish the connection between the dynamics of the inverse of U, and V, which characterizes the available inverse dynamics. We first normalize U to have unit diagonal. Next we remark that U has to inherit the reachability data of the original anticausal system. Hence, UW will be lower. Next, we establish that the $W^{-1}U^{-1}$ has to be upper, because it still contains the anticausal dynamics of the original inverse system. It follows that $VW^{-1}U^{-1}$ has to be causal, and hence that the reachability data of V has to match the observability data of $W^{-1}U^{-1}$, for which we can establish a precise expression, and this up to state equivalence. This produces all the data needed to determine U, with a simple formula.

IAS TUM Institute for Advanced Study

How it works...

First: if the factorization $T=LU$ exists, with U strictly outer, then also UW will be minimal lower. Normalizing U with $D_U=I$, we see that U must have a minimal realization of the form

$$U = I + F(I - Z^* A_a)^{-1} Z^* B_a$$

with so far unknown F. The whole problem reduces to finding F!

Formally computing the inverse $W^{-1}U^{-1}$ we find the upper

$$\begin{aligned} W^{-1}U^{-1} &= (D_W^* + B_W^*(I - Z^* A_W^*)^{-1} Z^* C_W^*) \cdot (I - F[I - Z^*(A_W^* - C_W^* F)]^{-1} Z^* C_W^*) \\ &= D_W^* + (B_W^* - D_W^* F)[I - Z^*(A_W^* - C_W^* F)]^{-1} Z^* C_W^* \end{aligned}$$

now, because L^{-1} has to be upper invertible, V has to bring this upper factor to lower from the left, hence the reachability pair of V must match the observability pair of $W^{-1}U^{-1}$, there must be a state transformation R such that

$$\begin{bmatrix} R A_V^* R^{-<-1>} \\ B_V^* R^{-<-1>} \end{bmatrix} = \begin{bmatrix} A_W^* & C_W^* \\ B_W^* & D_W^* \end{bmatrix} \begin{bmatrix} I \\ -F \end{bmatrix}$$

for $V(W^{-1}U^{-1}) = T_o^{-1} L$

upper

lower

the crucial property used...

In $$V(W^{-1}U^{-1}) = T_o^{-1}L$$

we see that V produces an external factorization on $W^{-1}U^{-1}$ *from the left.* This means that the reachability data for V is given by the observabilty data of $W^{-1}U^{-1}$*, in output normal form.* The observability data of $W^{-1}U^{-1}$ is:

$$\begin{bmatrix} A_W^* - C_W^* F \\ B_W^* - D_W^* F \end{bmatrix}$$

hence, we need a state transformation that puts this in output normal form, i.e., for some invertible R

$$\begin{bmatrix} A_V^* \\ B_V^* \end{bmatrix} = \begin{bmatrix} R^{-1}(A_W^* - C_W^* F)R^{<-1>} \\ (B_W^* - D_W^* F)R^{<-1>} \end{bmatrix}$$

and hence $$\begin{bmatrix} R A_V^* R^{-<-1>} \\ B_V^* R^{-<-1>} \end{bmatrix} = \begin{bmatrix} A_V^* & C_V^* \\ B_V^* & D_V^* \end{bmatrix} \begin{bmatrix} I \\ -F \end{bmatrix}$$ *which determines R and F.*

IAS TUM Institute for Advanced Study

the magic...

inverting the unitary factor produces the solution:

$$\begin{cases} R^{<-1>} = A_W R A_V^* + B_W B_V^* \\ F = -\left(C_W R A_V^* + D_W B_V^*\right) R^{-<-1>} \end{cases}$$

(warning: minimality of W is essential in the proof!)

Note: the two recursions (realization of *W* and O-I of *V*) are forward recursions, they allow for eventual re-ordering of the rows and columns of the matrix.

From the preceding, the left factor *L* is deduced straightforwardly:

$$\begin{aligned} L &= \left(D_u + C_u(I - ZA_u)^{-1} Z B_u\right) \cdot \left(D_W^* + B_V^*(I - Z^* A_V^*)^{-1} R^{-1} C_W^*\right) \\ &= \left(C_u Y R^{-1} C_W^* + D_u D_W^*\right) + C_u (I - ZA_u)^{-1} Z \left(A_u Y R^{-1} C_W^* + B_u D_W^*\right) \end{aligned}$$

but... it could also be gotten independently, by a dual algorithm.

Example

$$T = \begin{bmatrix} \boxed{1} & b_0 & & \\ a_0 & 1 & b_1 & \\ & a_1 & 1 & \ddots \\ & & \ddots & \ddots \end{bmatrix} \qquad T_u := TW = \begin{bmatrix} 1 & \boxed{b_0} & & \\ a_0 & 1 & b_1 & \\ & a_1 & 1 & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

$$T_u \sim \operatorname{diag}\left(\left[\begin{array}{c|c} & 1 \\ \cline{2-2} \cdot & \end{array}\right], \boxed{\left[\begin{array}{c|c} 1 & 0 \\ 0 & 0 \\ \hline 1 & b_0 \end{array}\right]}, \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline a_0 & 1 & b_1 \end{array}\right], \cdots \right)$$

Example (2)

Step -1:

$$\begin{bmatrix} | & 1 \\ \cdot & - \end{bmatrix} \begin{bmatrix} - & \cdot \\ 1 & | \end{bmatrix} = \begin{bmatrix} 1 & | \\ - & \cdot \end{bmatrix}$$

Step 0:

$$\left[\begin{array}{c|c} 1 & 0 \\ 0 & 1 \\ \hline 1 & b_0 \end{array}\right] \cdot \frac{1}{n_0} \left[\begin{array}{c|c} -b_0 & 1 \\ \hline 1 & b_0 \end{array}\right] = \frac{1}{n_0} \left[\begin{array}{c|c} -b_0 & 1 \\ 1 & b_0 \\ \hline 0 & n_0^2 \end{array}\right]$$

Step 1:

$$\left[\begin{array}{c|c} \frac{1}{n_0} & 0 \\ 0 & 1 \\ \hline \frac{1-a_0 b_0}{n_0} & b_1 \end{array}\right] \frac{1}{n_1} \left[\begin{array}{c|c} -b_1 & \frac{1-a_0 b_0}{n_0} \\ \hline \frac{1-a_0 b_0}{n_0} & b_1 \end{array}\right] = \frac{1}{n_1} \left[\begin{array}{c|c} -\frac{b_1}{n_0} & \frac{1-a_0 b_0}{n_0^2} \\ \frac{1-a_0 b_0}{n_0} & b_1 \\ \hline 0 & n_1^2 \end{array}\right]$$

IAS TUM Institute for Advanced Study

Example (3)

General step:

$$\left[\begin{array}{c|c} Y_{k,2} & 0 \\ 0 & 1 \\ \hline y_k & b_k \end{array}\right] \frac{1}{n_k} \left[\begin{array}{c|c} -b_k & y_k \\ \hline y_k & b_k \end{array}\right] = \frac{1}{n_k} \left[\begin{array}{c|c} -Y_{k,2} b_k & Y_{k,2} y_k \\ y_k & b_k \\ \hline 0 & n_k^2 \end{array}\right]$$

A purely " square root recursion" = stable QR-Type orthogonal recursion! that will be generally valid and stable and produces the LU solution when it exists, plus criteria for it to exist, plus information on kernels and Fredholm index when it does not.

The subsequent pivots in this case are: $1 + a_k \frac{Y_{k,2}}{Y_{k,1}}$

(note: the algorithm will always proceed, whether there is a solution or not, subsequent partial Schur complements are faithfully computed!)

IAS TUM Institute for Advanced Study

Interpretation

the result shows that even in the time-varying case poles and zeros determine the factors

Further considerations

1. The LU-factorization, when it exists, can be simply obtained from two recursive RQ factorizations! These always exist, and existence conditions can easily be derived from them. The algorithms are absolutely numerically stable, stability can even be improved with SVD's.
2. The algorithm has the flavor of a "linearization": it actually computes the "numerator" and the "denominator" of the local pivots separately (the pivot is a local Schur complement, that gets recursively computed, it is actually the entry L_{kk}).
3. As with all outer-inner factorizations, there is an underlying Riccati equation, for which the matrix R is a special solution. As is so often the case, it does not make sense to solve this Riccati equation, as the solution is produced by a linear, numerically stable and well-conditioned square-root equation!!!
4. In case the factorization conditions are not satisfied, then the factor V and its dual for L provide information on the kernels of the operator and the Fredholm index (in particular also the existence of "doubly invariant spaces".)