Computing the Kalman form

Clément Pernet, Aude Rondepierre, Gilles Villard

Computing the Kalman form

Clément Pernet
LMC, Université Joseph Fourier
51, rue des Mathématiques BP 53 IMAG-LMC 38041 Grenoble, FRANCE
clement.pernet@imag.fr

Aude Rondepierre
LMC, Université Joseph Fourier
51, rue des Mathématiques BP 53 IMAG-LMC 38041 Grenoble, FRANCE
aude.rondepierre@imag.fr

Gilles Villard
CNRS, LIP, Ecole Normale Supérieure de Lyon
46, Allée d'Italie, 69364 Lyon Cedex 07 FRANCE
gilles.villard@ens-lyon.fr

February 7, 2006

Abstract

We present two algorithms for the computation of the Kalman form of a linear control system. The first one is based on the technique developed by Keller-Gehrig for the computation of the characteristic polynomial. The cost is a logarithmic number of matrix multiplications. To our knowledge, this improves the best previously known algebraic complexity by an order of magnitude. Then we also present a cubic algorithm proven to be more efficient in practice.

1 Introduction

This report is a continuation of a collaboration of the first two authors on the algorithmic similarities between the computation of the Kalman form and of the characteristic polynomial. This collaboration led to [3, Theorem 2]. We report here an improvement of this last result based on a remark by the third author.

For a definition of the Kalman form of a linear control system, see [5, Theorem 1].

In this report we show how to adapt the branching algorithm of Keller-Gehrig [7, §5] (computing the characteristic polynomial) to compute the Kalman form. This implies an algebraic time complexity of $\mathcal{O}(n^\omega \log n)$. Now, the discussion of [2, §2] shows that a cubic algorithm, `LUK`, is more efficient in practice for the computation of the characteristic polynomial. Therefore, we adapt it to the computation of the Kalman form.

The outline of this report is the following : in section 2 we define the compressed Kyrlov matrix. It will help to describe the adaptation of Keller-Gehrig's algorithm to the computation of the Kalman form. In section 3, we recall Keller-Gehrig's algorithm. Section 4 presents the main result of this report, on the time complexity of the computation of the Kalman form. Lastly, we give a full description two algorithms to compute the Kalman form. The first one precises the operations used in section 4 to achieve the complexity and improves the constant hidden in the $\mathcal{O}()$ by saving operations. The second is based on another algorithm for the characteristic polynomial, that does not achieve the same algebraic complexity, but appears to be faster in practice.

We will denote by ω the exponent in the complexity of the matrix multiplication.

2 The compressed Krylov matrix

Let A and B be two matrices of dimension respectively $n \times n$ and $n \times m$. Consider the $n \times (mn)$ Krylov matrix K generated by the m column vectors of B and their iterates with the matrix A :

$$K = \left[\; b_1 \;\middle|\; \ldots \;\middle|\; A^{n-1}b_1 \;\middle|\; \ldots \;\middle|\; b_m \;\middle|\; \ldots \;\middle|\; A^{n-1}b_m \; \right]$$

Let r be the rank of K. $r \geq \text{rank}(B)$. Let us form the $n \times r$ non-singular matrix $\overline{K}$ by picking the first r linearly independent columns of K.

Definition 2.1. *$\overline{K}$ is the compressed Krylov matrix of B relatively to A.*

If a column vector $A^k b_j$ is linearly dependent with the previous column vectors, then any vector $A^l b_j, l > k$ will also be linearly dependent. Consequently the matrix $\overline{K}$ has the form :

$$\overline{K} = \left[\; b_1 \;\middle|\; \ldots \;\middle|\; A^{d_1-1}b_1 \;\middle|\; \ldots \;\middle|\; b_m \;\middle|\; \ldots \;\middle|\; A^{d_m-1}b_m \; \right] \tag{1}$$

for some d_i such that $0 \leq d_i \leq n-1$ and $\sum_{i=1}^{m} d_i = r$.

The order in the choice of the independent column vectors (from the left to the right) can also be interpreted in terms of lexicographical order on the sequence (d_i). Following Storjohann [8], we can therefore also define the compressed Krylov matrix as follows :

Definition 2.2. *The compressed Krylov matrix of B relatively to A is a matrix of the form*

$$\left[\; b_1 \;\middle|\; \ldots \;\middle|\; A^{d_1-1}b_1 \;\middle|\; \ldots \;\middle|\; b_m \;\middle|\; \ldots \;\middle|\; A^{d_m-1}b_m \; \right]$$

of rank r, such that the sequence (d_i) is lexicographically maximal.

The next section will present an algorithm to compute this compressed Krylov matrix.

3 Keller-Gehrig's algorithm

The selection of the linearly independent columns, starting from left to right, can be done by a gaussian elimination. A block elimination is mandatory to reduce the algebraic complexity to matrix multiplication. For this task, Keller-Gehrig first introduced in [7, §4] an algorithm called "step form elimination". The more recent litterature replaced it by the row echelon elimination (for example in [1]). We showed in [2] that the `LQUP` elimination (defined in [4]) of $\overline{K}^T$ could also be used (algorithm 3.1). This last algorithm simply returns the submatrix formed by the first independent column vectors of the input matrix form left to right.

Algorithm 3.1 `ColReducedForm`

Require: A a $m \times n$ matrix of rank r ($m, n \geq r$) over a field
Ensure: A' a $m \times r$ matrix formed by r linearly independent columns of A
1: $(L, Q, U, P, r) = \text{LQUP}(A^T)$ ($r = rank(A)$)
2: return $([I_r 0](Q^T A^T))^T$

Thus a straightforward algorithm to compute $\overline{K}$ would be to run algorithm 3.1 on the matrix K. But cost of the computation of K is prohibitive (n^3 coefficients and $\mathcal{O}(n^4)$ arithmetic operations with standard matrix product). Hence, the elimination process must be combined within the building of the matrix.

The computation of the iterates can rely on matrix multiplication, by computing the $\lceil \log_2(n) \rceil$ following powers of A :

$$A, A^2, \ldots, A^{2^i}, A^{2^{\lceil \log_2(n) \rceil - 1}}$$

Thus the following scheme,

$$\begin{cases} V_0 & = & [b_j] \\ V_{i+1} & = & [V_i | A^{2^i} V_i] \end{cases} \quad (2)$$

where the matrix V_i has 2^i columns, computes every iterates of b_j in $\mathcal{O}(n^\omega \log n)$ operations.

One elimination is performed after each application of A^{2^i}, to discard the linearly dependent iterates for the next iteration step. Moreover if a vector b_j has only $k < 2^i$ linearly independent iterates, one can stop the computation of its iterates. Therefore, the scheme (2) will only be applied on the block iterates of size 2^i.

From these remarks, we can now present Keller-Gehrig's algorithm. Although is was initially designed for the computation of the characteristic polynomial, we prefer to show it in a more general setting : the computation of the compressed Krylov matrix. Afterwards, we will show that the computation of the characteristic polynomial is a specialization of this algorithm with $B = I_n$ and that the recover of its coefficients is straightforward.

Algorithm 3.2 Compressed Krylov Matrix [Keller-Gehrig]

Require: A a $n \times n$ matrix over a field, B, a $n \times m$ matrix
Ensure: $(\overline{K}, r)$ as in (1)

$i = 0$
$V_0 = B = (V_{0,1}, V_{0,2}, \ldots, V_{0,m})$
$C = A$
while ($\exists k, V_k$ has 2^i columns) **do**
  for all j **do**
    if ($V_{i,j}$ has strictly less than 2^i columns) **then**
      $W_j = V_{i,j}$
    else
      $W_j = [V_{i,j} | C V_{i,j}]$
    end if
  end for
  $W = [W_1 | \ldots | W_n]$
  V_{i+1} = ColReducedForm(W) remember $r = rank(W)$ $\{V_{i+1} = [V_{i+1,1} | \ldots | V_{i+1,n}]$ where $V_{i+1,j}$ are the remaining vectors of W_j in $V_{i+1}\}$
  $C = C \times C$
  $i = i + 1$
end while
return (V_i, r)

Theorem 3.1 (Keller-Gehrig). *Suppose $m = \mathcal{O}(n)$. The compressed Krylov matrix of B relatively to A can be computed in $\mathcal{O}(n^\omega \log n)$ field operations.*

Proof. Algorithm 3.2 satisfies the statement (cf [7]). □

Property 3.1. *Let $\overline{K}$ be the compressed Krylov matrix of the identity matix relatively to A. The matrix $\overline{K}^{-1} A \overline{K}$ has the Hessenberg polycyclic form : it is block upper triangular, with companion*

blocks on its diagonal, and the upper blocks are zero except on their last column.

$$\overline{K}^{-1} A \overline{K} = \begin{bmatrix} \boxed{\begin{matrix} 0 & & & * \\ 1 & 0 & & * \\ & \ddots & \ddots & * \\ & & 1 & * \end{matrix}} & & & \\ & \ddots & & \\ & & & \begin{matrix} * \\ * \\ * \\ * \end{matrix} \\ & & \boxed{\begin{matrix} 0 & & & * \\ 1 & 0 & & * \\ & \ddots & \ddots & * \\ & & 1 & * \end{matrix}} & \end{bmatrix} \tag{3}$$

Corollary 3.1 (Keller-Gehrig). *The characteristic polynomial of A can be computed in $\mathcal{O}(n^\omega log n)$ field operation.*

Proof. The characteristic polynomial of the shifted Hessenberg form (3) is the product of the polynomials associated to the companion blocks on its diagonal. And since determinants are invariants under similarity transformations, it equals the characteristic polynomial of A. □

4 Computation of the Kalman form

Theorem 4.1 recalls the definition of the Kalman form of two matrices A and B.

Theorem 4.1. *Let A and B be two matrices of dimension respectively $n \times n$ and $n \times m$. Let r be the dimension of Span$(B, AB, \ldots, A^{n-1}B)$. There exist a non singular matrix T of dimension $n \times n$ such that*

$$T^{-1}AT = \begin{bmatrix} H & X \\ 0 & Y \end{bmatrix}, \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = T^{-1}B$$

where H and B_1 are respectively $r \times r$ et $r \times m$.

The main result of this report is the following result, based on an idea by the third author.

Theorem 4.2. *Let V be compressed Krylov matrix of B respectively to A. Complete V into a basis T of K^n by adding $n - r$ columns at the end of V. Then T satisfies the definition of the Kalman form of A and B.*

Proof. The matrix V satisfy the relation

$$AV = VH$$

where H is $r \times r$ and has the Hessenberg polycylic form (3). Let us note $T = [V|W]$.

Now

$$\begin{aligned} AT &= \left[\begin{array}{c|c} AV & AW \end{array}\right] \\ &= T \begin{bmatrix} H & X \\ 0 & Y \end{bmatrix}. \end{aligned}$$

Lastly, V is a basis of Span$(B, AB, \ldots, A^n B)$. Therefore each column of B is a linear combination of the colmuns of V :

$$B = T \begin{bmatrix} B_1 \\ 0 \end{bmatrix}.$$

□

Corollary 4.1. *The Kalman form of A and B can be computed in $\mathcal{O}(n^\omega log n)$.*

Proof. Applying theorem 3.1, there only remains to show how to complete V into T in $\mathcal{O}(n^\omega \log n)$. The idea is to complete V in its triangularized form. One computes the LUP factorization of V^T :

$$V^T = [L][\ U_1 \quad U_2\ ]P$$

Then replace $[U_1 U_2]$ by $\begin{bmatrix} U_1 & U_2 \\ 0 & Id \end{bmatrix}$ and $[L]$ by $\begin{bmatrix} L & 0 \\ 0 & Id \end{bmatrix}$ to get a $n \times n$ non singular matrix. This simply corresponds to set

$$T = \left[\begin{array}{c|c} \overline{K} & P^T \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} \end{array}\right].$$

It only costs $\mathcal{O}(n^\omega)$ field operations to recover the whole Kalman form (blocks H, X, Y and B_1), using for example matrix multiplications and matrix inversions. See section 5.2 for more details. □

This last result improves the algebraic time complexity for the computation of the Kalman form given in [3, Theorem 2] by an order of magnitude.

5 Algorithms into practice

The goal of the previous section was to establish the time complexity estimate and we therefore only sketched the algorithms involved. We will now focus more precisely on the operations so as to reduce the consant hiden in the $\mathcal{O}()$ notation.

5.1 Improvements on Keller-Gehrig's algorithm

The first improvement concerns the recover of the Hessenberg polycyclic form 3, once the compressed Krylov matrix is computed. In [7] Keller-Gehrig simply suggests to compute the product $K^{-1}AK$. This implies $4.66n^3$ additional field operation. We propose here to reduce this cost to ϕn^2, where ϕ is the number of blocks in the Hessenberg form. This technique was presented in [2]. We recall and extend it here for the recovery of the whole Hessenberg polycylic form.

First consider the case where the n first iterates of only one vector v are linearly independent.

Let $K = [v|Av|\dots|A^n v]$. The last column is the first which is linearly dependent with the previous. Let $P(X) = X^n - \sum_{i=0}^{n-1} m_i X^i$ represent this dependency (the minimal polynomial of this vector relatively to A). Again consider the LUP factorization of K^T. Let X_{n+1} denote the $n+1$th row of the matrix X and $X_{1\dots n}$ be the block of the first n rows of X. Then we have

$$K_{n+1}^T = (A^n v)^T = (\sum_{i=0}^{n-1} m_i A^i v)^T = [\ m_0 \quad \dots \quad m_{n-1}\ ](K^T)_{1\dots n}$$

Therefore

$$L_{n+1} = [\ m_0 \quad \dots \quad m_{n-1}\ ]L_{1\dots n}.$$

And the coefficients m_i can be recovered as the solution of a triangular system.

Now, one easily check that

$$K_{1\dots n}^{-1} A K_{1\dots n} = \begin{bmatrix} 0 & & & m_0 \\ 1 & 0 & & m_1 \\ & \ddots & \ddots & \vdots \\ & & 1 & m_{n-1} \end{bmatrix}.$$

This companion matrix is the Hessenberg polycyclic matrix to be computed.

In the situation of Keller-Gehrig's algorithm, the linear depencies also involve iterates of other vectors. However, the LQUP factorization will play a similar role than the previous LUP and makes it possible to recover the whole vector coefficients of the linear dependency.

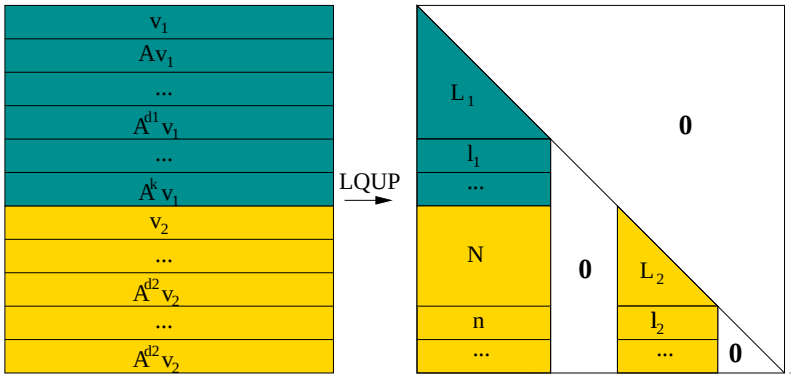

Figure 1: LQUP factorization of 2 blocks of iterates

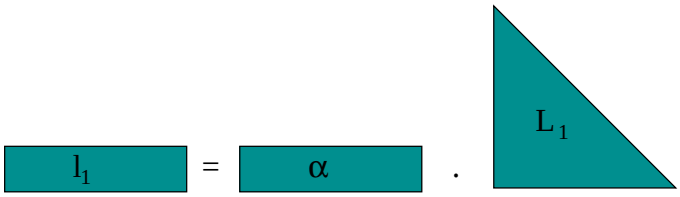

Figure 2: Recover of the coefficients of the first linear dependency

We show in figure 1 the case of two blocks of iterates. The first linear dependency relation (for $A^{d_1}v_1$) is done as previously (see figure 2).

Now for the second block, the first linearly dependent vector $A^{d_2}v_2$ satisfies a relation of the type :

$$A^{d_2}v_2 = \sum_{i=0}^{d_2-1} \beta_i A^i v_2 + \sum_{i=0}^{d_1-1} \gamma_i A^i v_1$$

The vector of coefficients $\beta = [\beta_i]$ and $\gamma = [\gamma_i]$ can be obtained by solving the following system shown in figure 3.

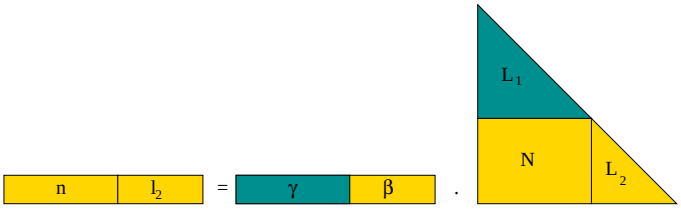

Figure 3: Recover of the coefficients of the second linear dependency

There only remains to build the Hessenberg polycyclic matrix from these vectors :

$$H = \begin{bmatrix} 0 & & & \alpha_0 & & & \gamma_0 \\ 1 & 0 & & \alpha_1 & & & \gamma_1 \\ & \ddots & \ddots & \vdots & & & \vdots \\ & & 1 & \alpha_{d_1-1} & & & \gamma_{d_1-1} \\ & & & & 0 & & \beta_0 \\ & & & & 1 & 0 & \beta_1 \\ & & & & & \ddots & \ddots & \vdots \\ & & & & & & 1 & \beta_{d_2-1} \end{bmatrix}.$$

This technique can be applied to every block of iterates. Therefore the Hessenberg polycyclic matrix can be recovered by as many triangular system resolutions as the number of blocks.

5.2 The main algorithm

We have seen in section 4 how to compute the matrix T (simply $T = \left[\begin{array}{c|c} \overline{K} & P^T \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} \end{array}\right]$). Section 5.1 showed how to compute H. There only remains to show how to compute the matrices X, Y and B_1 and we will be done.

We recall (from the proof of theorem 4.2) that

$$\begin{aligned} AT &= \left[\begin{array}{c|c} AV & AP^T \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} \end{array}\right] \\ &= T \begin{bmatrix} H & X \\ 0 & Y \end{bmatrix} \end{aligned}$$

Then X and Y satisfy $T \begin{bmatrix} X \\ Y \end{bmatrix} = AP^T \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix}$. Let us write

$$A' = PAP^T = \begin{bmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{bmatrix}.$$

We have

$$T \begin{bmatrix} X \\ Y \end{bmatrix} = P^T A' \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} = P^T \begin{bmatrix} A'_{12} \\ A'_{22} \end{bmatrix}$$

Now

$$T = P^T \begin{bmatrix} U_1^T & 0 \\ U_2^T & I_{n-r} \end{bmatrix} \begin{bmatrix} L^T & 0 \\ 0 & I_{n-r} \end{bmatrix},$$

therefore

$$\begin{aligned} \begin{bmatrix} U_1^T & 0 \\ U_2 & I_{n-r} \end{bmatrix} \begin{bmatrix} L^T & 0 \\ 0 & I_{n-r} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} &= \begin{bmatrix} A'_{12} \\ A'_{22} \end{bmatrix} \\ \begin{bmatrix} U_1^T & 0 \\ U_2^T & I_{n-r} \end{bmatrix} \begin{bmatrix} L^T X \\ Y \end{bmatrix} &= \begin{bmatrix} A'_{12} \\ A'_{22} \end{bmatrix} \end{aligned}$$

And the system

$$\begin{cases} U_1^T L^T X & = & A'_{12} \\ U_2^T L^T X + Y & = & A'_{22} \end{cases}$$

has the following solution

$$\begin{cases} X & = & L^{-T} U_1^{-T} A'_{12} \\ Y & = & A'_{22} - U_2^T U_1^{-T} A'_{12} \end{cases}$$

The computation of B_1 is straightforward from the equation $TB_1 = B$:

$$B_1 = L^{-T}U_1^{-T}PB.$$

We are now able to write the algorithm.

Algorithm 5.1 Kalman form

Require: A a $n \times n$ matrix over a field, B, a $n \times m$ matrix
Ensure: r, T, H, X, Y, B_1 as in theorem 4.1
1: $(V, r) =$ `CompressedKrylovMatrix`(A, B)
2: **if** (r=n) **then**
3: return $(n, Id, A, \emptyset, \emptyset, B)$
4: **else**
5: $(L, [U_1 U_2], P) =$ LUP(V^T)
6: $T = \left[\begin{array}{c|c} V & P^T \begin{bmatrix} 0 \\ I_{n-r} \end{bmatrix} \end{array}\right]$
7: $B_1 = L^{-T}U_1^{-T}PB$
8: $A' = PAP^T = \begin{bmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{bmatrix}$
9: $X = L^{-T}U_1^{-T}A'_{12}$
10: $Y = A'_{22} - U_2^T U_1^{-T} A'_{12}$
11: **for all** j **do**
12: $m_j = l_j L_j^{-1}$ as explained in section 5.1
13: **end for**
14: Build the polycyclic matrix H using the m_j as shown in section 5.1.
15: return (r, T, H, X, Y, B_1)
16: **end if**

Lastly, note that the LUP factorization of V^T is already computed at the end of the call to `CompressedKrylovMatrix`. Thus, step 5 in algorithm 5.1 can be skipped.

5.3 LU-Krylov : a cubic variant

In [2], we introduce an algorithm for the computation of the characteristic polynomial : `LUK`. Alike Keller-Gehrig's algorithm, it is also based on the Krylov iterates of several vectors and relies as much as possible on matrix multiplication. But the krylov iterates are computed with matrix vector products, so as to avoid the log$n$ factor in the time complexity. As a consequence it is $\mathcal{O}(n^3)$ algorithm, but we showed that it was faster in practice.

Algorithm 5.2 shows how to adapt this algorithm to the computation of the Kalman form. We expect this algorithm to be the more efficient in practice.

References

[1] Michael Clausen, Peter Burgisser, and Mohammad A. Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[2] Jean-Guillaume Dumas, Clément Pernet, and Zhendong Wan. Efficient computation of the characteristic polynomial. In Kauers [6].

[3] Jean-Guillaume Dumas and Aude Rondepierre. Algorithms for symbolic/numeric control of affine dynamical system. In Kauers [6].

[4] Oscar H. Ibarra, Shlomo Moran, and Roger Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *Journal of Algorithms*, 3(1):45–56, March 1982.

Algorithm 5.2 `Kalman-LUK` : Kalman-LU-Krylov

Require: A a $n \times n$ matrix over a field, B, a $n \times m$ matrix

Ensure: r, T, H, X, Y, B_1 as in theorem 4.2

$v = B_1$

$\begin{cases} K = \begin{bmatrix} v & Av & A^2v & \dots \end{bmatrix} \\ (L, [U_1 | U_2], P) = \texttt{LUP}(K^T), r_1 = \text{rank}(K) \end{cases}$ {The matrix K is computed on the fly : at most $2r_1$ columns are computed}

$m = (m_1, \dots, m_{r1}) = L_{r_1+1} L_{1\dots r_1}^{-1}$

$f = X^{r_1} - \sum_{i=1}^{r_1} m_i X^{i-1}$

if $(r_1 = n)$ **then**

  return $(n, Id, A, \emptyset, \emptyset, B)$

else

  $A' = PAP^T = \begin{bmatrix} A'_{11} & A'_{12} \\ A'_{21} & A'_{22} \end{bmatrix}$ where A'_{11} is $r_1 \times r_1$.

  $A_R = A'_{22} - U_2^T U_1^{-T} A'_{12}$

  $B' = \begin{bmatrix} L^{-T} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} U_1^{-T} & 0 \\ -U_2^T U_1^{-T} & I \end{bmatrix} PB$

  Compute the permutation Q s.t. $B'Q = \begin{bmatrix} * & * \\ 0 & Z \end{bmatrix}$ $\{Z$ is $(n - r_1) \times \mu\}$

  if $(\mu = 0)$ **then**

    $X = L^{-T} U_1^{-T} A'_{12}$

    $Y = A_R$

    $T = \left[\begin{array}{c|c} K & P^T \begin{bmatrix} 0 \\ I_{n-r_1} \end{bmatrix} \end{array} \right]$

    return (r_1, T, C_f, X, Y, X)

  else

    $(r_2, T^{(2)}, H^{(2)}, X^{(2)}, Y^{(2)}, B_1^{(2)}) = \texttt{Kalman-LUK}(A_R, Z)$

    $T = \left[\begin{array}{c|c} K & P^T \begin{bmatrix} 0 \\ T^{(2)} \end{bmatrix} \end{array} \right]$

    $J = L^{-T} U_1^{-T} A'_{12} T^{(2)} = \begin{bmatrix} J_1 & J_2 \end{bmatrix}$ $\{J_1$ is $r_1 \times r_2$ and J_2, $r_1 \times (n - r_1 - r_2)\}$

    $H = \begin{bmatrix} C_f & J_1 \\ 0 & H^{(2)} \end{bmatrix}$

    $X = \begin{bmatrix} J_2 \\ X^{(2)} \end{bmatrix}$

    return $(r_1 + r_2, T, H, X, Y^{(2)}, B_1)$

  end if

end if

[5] R.E. Kalman. Canonical structure of linear dynamical systems. In *Proceedings of the National Academy of Sciences*, pages 596–600, 1961.

[6] Manuel Kauers, editor. *ISSAC'2005. Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, Beijing, China*. ACM Press, New York, July 2005.

[7] Walter Keller-Gehrig. Fast algorithms for the characteristic polynomial. *Theoretical computer science*, 36:309–317, 1985.

[8] Arne Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Institut für Wissenschaftliches Rechnen, ETH-Zentrum, Zürich, Switzerland, November 2000.