Time-Varying Systems and Computations

Klaus Diepold, LDV

November 30, 2011

Homework 2

Task Description

The homework provides problems to determine and check elementary properties of time-varying systems The goal is to check some important identities by manual computations. A further goal is to program simple Matlab functions, which will be useful for future computations in the course.

Date of delivery: **November 28, 2011, 10:00 o'clock**
Send the deliverable via email to **tvsc@ldv.ei.tum.de.**

1 Observability and Controllability Gramians

Problem

Recall the definition of the time-varying observability matrix $\mathcal{O}_k$ and the time-varying controllability matrix $\mathcal{C}_k$

$$\mathcal{C}_k = \begin{bmatrix} B_{k-1} & A_{k-1}B_{k-2} & A_{k-1}A_{k-2}B_{k-3} & \cdots \end{bmatrix}, \quad \mathcal{O}_k = \begin{bmatrix} C_k \\ C_{k+1}A_k \\ C_{k+2}A_{k+1}A_k \\ \vdots \end{bmatrix}.$$

1. Determine a recursive formula to compute the sequence of Observability Gramians, defined as $K_k = \mathcal{O}_k^T \mathcal{O}_k$.
2. Determine a recursive formula to compute the sequence of Controllability Gramians, defined as $W_k = \mathcal{C}_k \mathcal{C}_k^T$

Solution

1. We calculate the Observability Gramian matrix according to

$$\mathcal{O}_k^T \mathcal{O}_k = \begin{bmatrix} C_k^T & A_k^T C_{k+1}^T & A_k^T A_{k+1}^T C_{k+2} & \cdots \end{bmatrix} \begin{bmatrix} C_k \\ C_{k+1}A_k \\ C_{k+2}A_{k+1}A_k \\ \vdots \end{bmatrix} =$$

$$= C_k^T C_k + A_k^T C_{k+1}^T C_{k+1} A_k + A_k^T A_{k+1}^T C_{k+2} C_{k+2} A_{k+1} A_k + \cdots .$$

Using the shorthand $K_k = \mathcal{O}_k^T \mathcal{O}_k \geq 0$ for the Observability Gramian we can re-write the lengthy expression recursively as

$$K_k = C_k^T C_k + A_k^T K_{k+1} A_k.$$

This equation is commonly referred to as a Lyapunov equation. The dynamic degree of a minimal state-space realization corresponds with the rank of the observability Gramian, i.e. with the number or linear independent rows of the observability matrix. We can compute this rank and the associated basis for the space by applying an SVD to the Hankel matrix.

We know from control engineering that a system is called completely observable, if the observability matrices $\mathcal{O}_k$ have full row rank, or, equivalently that all observability Gramians $\mathcal{K}_k$ are positive definite.

2. We calculate the Controllability Gramian matrix

$$K_k = \mathcal{C}_k\mathcal{C}_k^T = \begin{bmatrix} B_{k-1} & A_{k-1}B_{k-2} & A_{k-1}A_{k-2}B_{k-3} & \cdots \end{bmatrix} \begin{bmatrix} B_{k-1}^T \\ B_{k-2}^T A_{k-1}^T \\ B_{k-3}^T A_{k-2}^T A_{k-1}^T \\ \cdots \end{bmatrix}$$

$$= B_{k-1}B_{k-1}^T + A_{k-1}B_{k-2}B_{k-2}^T A_{k-1}^T + A_{k-1}A_{k-2}B_{k-3}B_{k-3}^T A_{k-2}^T A_{k-1}^T + \cdots .$$

Using the shorthand $W_k = \mathcal{C}_k\mathcal{C}_k^T \geq 0$ for the Controllability Gramian we can re-write the lengthy expression recursively as

$$W_k = B_{k-1}B_{k-1}^T + A_{k-1}^T W_{k-1}A_{k-1}.$$

This equation is commonly referred to as a Lyapunov equation. The dynamic degree of a minimal state-space realization corresponds with the rank of the controllability Gramian, i.e. with the number or linear independent columns of the controllability matrix. We can compute this rank and the associated basis for the space by applying an SVD to the Hankel matrix.

We know from control engineering that a system is called completely controllable, if the controllability matrices $\mathcal{C}_k$ have full row rank, or, equivalently that all controllability Gramians $\mathcal{W}_k$ are positive definite.

2 State-Transformation

Problem

Recall that we can apply a linear, non-singular transformation to the state-space

$$R_k x_k = x_k', \quad \det R_k \neq 0, \quad \forall k.$$

You should know that this transformation leaves the transfer operator T invariant.

1. Determine algebraic expressions describing the effects of a state-transformation on the Controllability matrix $\mathcal{C}_k$ and the Controllability Gramian W_k.

2. Determine algebraic expressions describing the effects of a state-transformation on the Observability matrix $\mathcal{O}_k$ and the Observabilty Gramian K_k.

Solution

1. State Transformation induces the identity for the realization matrix

$$\begin{bmatrix} A_k' & B_k' \\ C_k' & D_k' \end{bmatrix} = \begin{bmatrix} R_{k+1}A_kR_k^{-1} & R_{k+1}B_k \\ C_kR_k^{-1} & D_k \end{bmatrix}.$$

The effect of state-transformation translates to the controllability matrices

$$\begin{aligned} \mathcal{C}_k' &= \begin{bmatrix} B_{k-1}' & A_{k-1}'B_{k-2}' & A_{k-1}'A_{k-2}'B_{k-3}' & \cdots \end{bmatrix} \qquad (1) \\ &= \begin{bmatrix} (R_kB_{k-1}) & (R_kA_{k-1}R_{k-1}^{-1})(R_{k-1}B_{k-2}) & (R_kA_{k-1}R_{k-1}^{-1})(R_{k-1}A_{k-2}R_{k-2}^{-1})(R_{k-2}B_{k-3}) & \cdots \end{bmatrix} \\ &= R_k \begin{bmatrix} B_{k-1} & A_{k-1}B_{k-2} & A_{k-1}A_{k-2}B_{k-3}) & \cdots \end{bmatrix} \\ &= R_k\mathcal{C}_k. \end{aligned}$$

Based on this result we can determine the controllability Gramian for the transformed system

$$W'_k = \mathcal{C}'_k \mathcal{C}'^T_k = R_k \mathcal{C}_k \mathcal{C}^T_k R^T_k = R_k W_k R^T_k,$$

which is a congruence transformation of the controllability Gramian. While the congruence transformation modifies the eigenvalues it leaves the signs of the eigenvalues invariant.

2. The corresponding effect on the observability matrix is

$$\begin{aligned}
\mathcal{O}'_k &= \begin{bmatrix} C'_k \\ C'_{k+1}A'_k \\ C'_{k+2}A'_{k+1}A'_k \\ \vdots \end{bmatrix} = \begin{bmatrix} C_k R_k^{-1} \\ C_{k+1}R_{k+1}^{-1}R_{k+1}A_k R_k^{-1} \\ C_{k+2}R_{k+2}^{-1}R_{k+2}A_{k+1}R_{k+1}^{-1}R_{k+1}A_k R_k^{-1} \\ \vdots \end{bmatrix} \qquad (2) \\
&= \begin{bmatrix} C_k R_k^{-1} \\ C_{k+1}A_k R_k^{-1} \\ C_{k+1}A_{k+1}A_k R_k^{-1} \\ \vdots \end{bmatrix} = \mathcal{O}_k R_k^{-1}.
\end{aligned}$$

Based on this result we can determine the observability Gramian for the transformed system as

$$K'_k = \mathcal{O}'^T_k \mathcal{O}'_k = R_k^{-T} \mathcal{O}^T_k \mathcal{O}_k R_k^{-1} = R_k^{-T} K_k R_k^{-1},$$

which is a congruence transformation of the controllability Gramian. While the congruence transformation modifies the eigenvalues it leaves the signs of the eigenvalues invariant.

For a number of optimization steps in the context of designing linear state-space systems one needs to compute the product of observability and controllability Gramians, id est

$$K'_k W'_k = (R_k^{-T} K_k R_k^{-1})(R_k W_k R_k^T) = R_k^{-1} K_k W_k R_k,$$

where the state-transformation induces a similarity transformation on the product of the observability and the controllability Gramians. Notice that a similarity transformation leaves the eigenvalues of $K_k W_k$ invariant.

3 State-Space Arithmetic

Problem

We have two lower triangular matrice T_1 and T_2. For each matrix we have a corresponding time-varying state-space realizations

$$T_1 \leftrightarrow \Sigma_1 = \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right], \quad T_2 \leftrightarrow \Sigma_2 = \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right].$$

In order to develop matrix algorithms that directly work in state-space we need to determine elementary arithmetic operations on matrices in state-space.

1. Determine an expression for a state-space realization Σ_I for the **inversion** of the matrix T_1, i.e. for $T_I = T_1^{-1}$ in terms of the state-space realizations for T_1.
2. Determine an expression for a possibly non-minimal state-space realization Σ_S for the **sum of the matrices** $T_S = T_1 + T_2$ in terms of the state-space realizations for T_1 and T_2 (see Figure 2).
3. Determine an expression for a possibly non-minimal state-space realization Σ_P for the **product of matrices** $T_P = T_2 \cdot T_1$ in terms of the state-space realizations for T_1 and T_2 (see Figure 3).
4. Determine an expression for a possibly non-minimal state-space realization Σ_F for the **feedback connection of the two matrices** T_1 and T_2 as is depicted in Figure 4

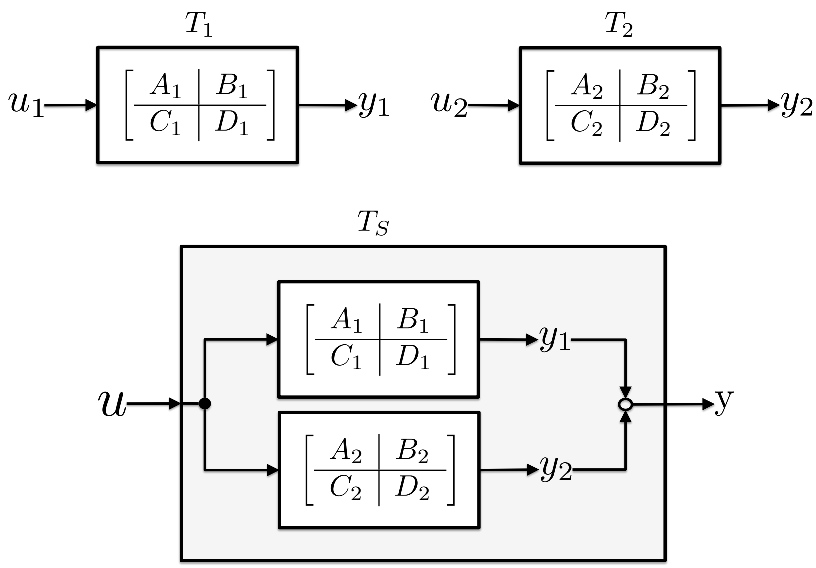

Figure 1: State-Space Realization for sum of two matrices

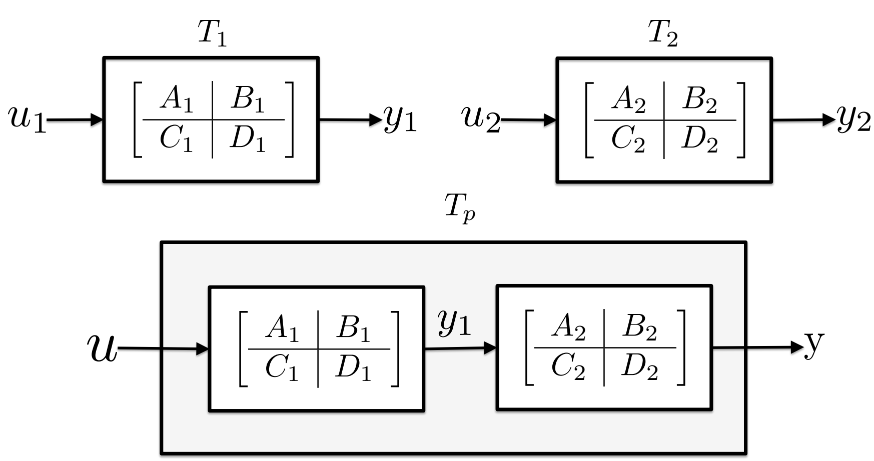

Figure 2: State-Space Realization for product of two matrices

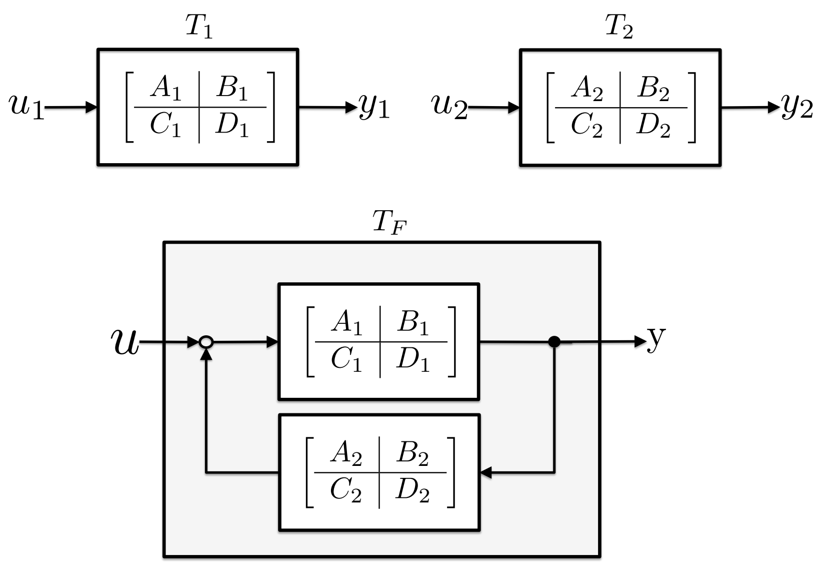

Figure 3: State-Space Realization for the feedback conncetion of two matrices

Solution

1. Considering the state equations of a linear time-variant system

$$x_{k+1} = A_k x_k + B_k u_k, \quad y_k = C_k x_k + D_k u_k$$

we just have to solve these equations to express the next state and the input signal as a linear function of the output signal and the previous state. We can identify the realization matrix for the inverse system

$$\Sigma_I = \left[\begin{array}{c|c} A_k - B_k D_k^{-1} C_k & B_k D_k^{-1} \\ \hline -D_k^{-1} C_k & D_k^{-1} \end{array}\right].$$

Notice that the realization matrix for the inverse system is not the inverse of the realization matrix for the original system.

2. For the sum of the two systems we can add the output equations Combining the equations into one block matrix notation and using the state-vector $x_k = \begin{bmatrix} x_k^{[1]} \\ x_k^{[2]} \end{bmatrix}$ we get the realization matrix

$$\Sigma_S = \left[\begin{array}{c|c} \begin{bmatrix} A_k^{[1]} & 0 \\ 0 & A_k^{[2]} \end{bmatrix} & \begin{bmatrix} B_k^{[1]} \\ B_k^{[2]} \end{bmatrix} \\ \hline \begin{bmatrix} C_k^{[1]} & C_k^{[2]} \end{bmatrix} & [D_k^{[1]} + D_k^{[2]}] \end{array}\right]$$

3. For the product of two matrices we can use the relations $u_k^{[1]} = u_k$, $u_k^{[2]} = y_k^{[1]}$ and $y_k = y_k^{[2]}$ to formulate the output equations according to Putting all equations together we can directly identify the realization matrix

$$\Sigma_P = \left[\begin{array}{cc|c} A_k^{[1]} & 0 & B_k^{[1]} \\ B_k^{[2]} C_k^{[1]} & A_k^{[2]} & B_k^{[2]} D_k^{[1]} \\ \hline D_k^{[2]} C_k^{[1]} & C_k^{[2]} & D_k^{[2]} D_k^{[1]} \end{array}\right]$$

4. The feedback connection of the two systems induces the identities $u_k^{[1]} = u_k + y_k^{[2]}$, $y_k = y_k^{[1]}$ and $u_k^{[2]} = y_k$ producing a Toeplitz operator of the form

$$T_F = (1 - T_1 T_2)^{-1} T_1.$$

The realization for this Toeplitz operator can be determined by straight forward but tedious calculations

$$\Sigma_F = \left[\begin{array}{cc|c} A_k^{[1]} & B_k^{[1]} C_k^{[2]} & B_k^{[1]} \\ 0 & A_k^{[2]} & 0 \\ \hline 0 & 0 & 0 \end{array}\right] + \begin{bmatrix} B_k^{[1]} D_k^{[2]} \\ B_k^{[2]} \\ 1 \end{bmatrix} (1 - D_k^{[1]} D_k^{[2]})^{-1} \begin{bmatrix} C_k^{[1]} & D_k^{[1]} C_k^{[2]} & D_k^{[1]} \end{bmatrix}.$$

4 Schur-Complement

Problem

We have a matrix M, which consists of four block-matrices as $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

1. Determine the matrix x such that the 21-block entry in M is eliminated according to the equation

$$\begin{bmatrix} 1 & \\ x & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & \Delta \end{bmatrix}.$$

2. If x is determined to eliminate the 21-block matrix in M what is the expression for the resulting matrix Δ? The expression for Δ is called a *Schur Complement.*

3. If you have the linear system of equation $Tu = y$. How do you have to map T, u and y onto the block matrices of M such that Γ contains the solution vector $u = T^{-1}y$? (Assume all neccessary inverses to exist).

4. Determine the matrix y such that the 12-block entry in $\begin{bmatrix} a & b \\ 0 & \Delta \end{bmatrix}$ is eliminated according to the equation

$$\begin{bmatrix} a & b \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} 1 & y \\ & 1 \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & \Delta \end{bmatrix}.$$

5. Using the results of the previous task we can write a factorization of the matrix M in the form

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & \\ x & 1 \end{bmatrix}^{-1} \begin{bmatrix} a & \\ & \Delta \end{bmatrix} \begin{bmatrix} 1 & y \\ & 1 \end{bmatrix}^{-1}.$$

Utilizing this factorization give a formula for the inverse M^{-1} in terms of the block-entries a, b, c, d and Δ.

6. Determine the entries of a matrix M such that the representation of the transfer operator $T = D + C\,(1 - ZA)^{-1}\,ZB$ is the Schur Complement of this M.

Solution

1. From the specification given we can read off the requirement for x

$$x \cdot a + c = 0 \qquad \Rightarrow x = -c \cdot a^{-1}.$$

2. Once we plug the result for x in the expression for Δ we see

$$\Delta = x \cdot b + d = d - c \cdot a^{-1} \cdot b,$$

which is the *Schur Complement* of M with regards to a denoted by M/a.

3. Taking a linear system of equations $Tu = y$ we can determine the solution vector u if we take M as

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} := \begin{bmatrix} T & y \\ -1 & 0 \end{bmatrix}$$

and eliminating the 21-block to generate the Schur Complement

$$d - c \cdot a^{-1} \cdot b = 0 - (-1) \cdot T^{-1} \cdot y = T^{-1} \cdot y = u$$

4. From the specification given we can read off the expression to determine the matrix y as

$$a \cdot y + b = 0 \qquad \Rightarrow y = -a^{-1} \cdot b.$$

5. Putting the results from the previous computations together we have

$$\begin{aligned} M &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & \\ -ca^{-1} & 1 \end{bmatrix}^{-1} \begin{bmatrix} a & \\ & \Delta \end{bmatrix} \begin{bmatrix} 1 & -a^{-1}b \\ & 1 \end{bmatrix}^{-1} \qquad (3) \\ &= \begin{bmatrix} 1 & \\ ca^{-1} & 1 \end{bmatrix}^{-1} \begin{bmatrix} a & \\ & \Delta \end{bmatrix} \begin{bmatrix} 1 & a^{-1}b \\ & 1 \end{bmatrix}^{-1} \end{aligned}$$

The inverse of M can then be expressed in terms of the block entries as

$$\begin{aligned} M^{-1} &= \begin{bmatrix} 1 & -a^{-1}b \\ & 1 \end{bmatrix}^{-1} \begin{bmatrix} a^{-1} & \\ & \Delta^{-1} \end{bmatrix} \begin{bmatrix} 1 & \\ -ca^{-1} & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} a^{-1} + a^{-1}b\Delta^{-1}ca^{-1} & -a^{-1}b\Delta^{-1} \\ \Delta^{-1}ca^{-1} & \Delta^{-1} \end{bmatrix} \end{aligned} \tag{4}$$

6. We can identify the entries of M

$$M = \begin{bmatrix} 1 - ZA & ZB \\ -C & D \end{bmatrix},$$

such that the Schur Complement of M/a produces the representation of the Toeplitz Operator T.

5 Hankel Singular Values

Problem

Write a Matlab program that returns the Hankel singular values of a lower triangular $n \times n$ matrix T, i.e. the program shall return the singular values of all Hankel matrices

$$\mathcal{H}_k = \begin{bmatrix} T_{k,k-1} & T_{k,k-2} & T_{k,k-3} & \cdots \\ T_{k+1,k-1} & T_{k+1,k-2} & T_{k+1,k-3} & \cdots \\ T_{k+2,k-1} & T_{k+2,k-2} & T_{k+2,k-3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad k = 1, 2, \ldots n.$$

The call to this function could look like

```
s = hankel_svd(T),
```

where T is the given $n \times n$ matrix and s is a $n \times n$ matrix, where the k-th column contains the singular values of $\mathcal{H}_k$.

Solution

```
function S = hankel_svd(T)
[rows, cols] = size(T);
S = zeros(cols);
for k = 1:cols
     sig=svd(fliplr(T(k:end,1:k-1)));
     S(1:length(sig),k) = sig;
end
```

6 QR-Factorization

Problem

1. Write a Matlab program that computes the QR-decomposition of a square matrix A using Givens rotations. The call into this function can look like

   ```
   [q,r] = givens_qr(A),
   ```

   where q denotes the orthonormal matrix Q and r the upper triangular matrix R in the factorization $A = QR$.

2. Compare and discuss the results produced by your function with the results produced with Matlab's built-in QR Decomposition.

3. Discuss if the QR decomposition of a square matrix A is unique.

Solution

1. Matlab Programs

```
% File: GivensQR.m
% Purpose: General: Compute QR decomposition of data matrix; For Least Squares Problems y = X*b + e:
% Compute QR decomposition of data matrix X and determine projection of data vector y onto column sp
% of Q on the fly.
%
% Method: Standard QR decomposition algorithm for computing X = Q*R, Q'*Q = 1, R upper triangular ba
% rotations, as explained in Golub/VanLoan: 'Matrix Computations'.
% Optionally, if a vector y is specified, the orthogonal projection of y onto the column space of Q,
% [y;r] = Q'*y is computed on the fly, where r represents the projected residual
%
% Matlab Call: [R,y,r] = GivensQR(X,y,`Q`)
% Input: X:  mxn real-valued data matrix
% y:  mxny regressand vector (optionally)
% 'Q' character to indicate if Q is required in factored form
%
% Output: R: triu(R): upper triangular Cholesky factor of X'*X = R'*R if option 'Q' is set, then
%     tril(R,-1) contains lower triangular matrix of single parameter representation of the orthogon
%     transformation Q in factored form
% y:  nxny projected regressand vector (optionally)
% r:  (m-n)xny vector of projected residual

function [R,y,r] = GivensQR(X,y,Q)
if nargin > 1,
   if nargin < 3,
      if isstr(y),
         Q = y;
         clear y;
      else
         Q = 'NQ';
      end
   end
else
  Q = 'NQ';
end

[m,n] = size(X);
ny = 0;
  if exist('y'),
    ny = size(y,2);
    X = [X,y];
  end
  for j=1:n,
     for i=m:-1:j+1;
         [cosinus,sinus] = Givens(X(i-1,j),X(i,j));
         X(i-1:i,j:n+ny) = RowRot(X(i-1:i,j:n+ny),cosinus,sinus);
 if strcmp(Q,'Q')
            X(i,j) = cosinus;
         end
      end
  end
  if strcmp(Q,'Q')
```

```
      R = X(:,1:n);
   else
      R = triu(X(1:min(m,n),1:n));
   end
   if nargout > 1,
      y = X(1:min(m,n),n+1:n+ny);
   end
   if nargout > 2 & (m >n),
      r = X(n+1:m,n+1:n+ny);
   end

% Purpose: Determine rotation parameters cosinus and sinus of orthogonal 2x2 Givens matrix
% such that
% [cosinus -sinus ] * [a] = [sqrt(a^2+b^2)]
% [sinus   cosinus]   [b]   [     0       ]
%
% Input: a,b:
% Output: cosinus: cosinus of rotation angle
% sinus:   sinus of rotation angle
%
% Matlab Call: [cosinus,sinus] = Givens(a,b)

function [cosinus,sinus] = Givens(a,b)

if b==0,
   cosinus = 1;
   sinus = 0;
elseif a==0,
   cosinus = 0;
   sinus = 1;
elseif a==0 & b==0
   cosinus = 1;
   sinus = 0;
else
   if abs(b) > abs(a)
      tangens = -a/b;
      sinus = sign(a)*1/sqrt(1+tangens^2);
      cosinus = sinus*tangens;
   else
      tangens = -b/a;
      cosinus = sign(a)*1/sqrt(1+tangens^2);
      sinus = cosinus*tangens;
   end
end

function [X,Y] = RowRot(X,Y,c,s)
% function [X,Y] = RowRot(X,Y,c,s)

    t1 = X;
    t2 = Y;
    X = c*t1 - s*t2;
    Y = s*t1 + c*t2;
```

2. This question may also be answered by checking if the QR decomposition is unique. Assume that

we have a QR decomposition for a given matrix A, id est, we have

$$A = QR, \quad A \in \mathbb{R}^{n \times n}$$

with $Q^T Q = 1$ and an upper triangular matrix R. The question for uniqueness amounts to checking if we can find alternative matrices Q' and R', which have the same properties and which represent a QR decomposition of the matrix A. To this end we insert the identity between the factors of the QR decomposition as

$$A = QR = \underbrace{QM^{-1}}_{Q'}\underbrace{MR}_{R'}.$$

For Q' to be a alternative orthogonal factor it must be an orthogonal matrix, which means that $M^{-1} = M^T$ must be satisfied. For R' to be an alternative factor we must have M to be an upper triangular matrix. Taken together, M must be upper triangular and orthogonal, a requirement which is only satisfied by diagonal matrices consisting of ± 1 on its diagonal. That is, the QR decomposition computed by Matlab may only differ from the result computed by our own implementation by signs.