Technische Universität Berlin **Sommersemester 2012**
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik
Tobias Brüll

Systems and control theory

Series 12

Task 1:
Let $A, Q = Q^* \in \mathbb{C}^{n,n}$, $B, S \in \mathbb{C}^{n,m}$, and $R = R^* \in \mathbb{C}^{m,m}$. Then $X = X^* \in \mathbb{C}^{n,n}$ is called a solution of the *algebraic Riccati equation*, if we have

$$0 = Q - A^*X - X^*A - (S - X^*B)R^{-1}(S^* - B^*X). \tag{1}$$

(a) Rewrite (1) in the form

$$X^*GX + F^*X + X^*F - H = 0, \tag{2}$$

with $F, G = G^*, H = H^* \in \mathbb{C}^{n,n}$. This equation is also called algebraic Riccati equation.

(b) Define the Hamiltonian matrix (cf. Series 11, Task 7) $\mathcal{H} := \begin{bmatrix} F & G \\ H & -F^* \end{bmatrix}$.
Show that $X \in \mathbb{C}^{n,n}$ solves (2) if and only if

$$\mathcal{H} \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X^* \end{bmatrix} (F + GX). \tag{3}$$

(c) Conclude that if $X = X^*$ is Hermitian and fulfills (3) then every eigenvalue of $(F + GX)$ is also an eigenvalue of $\mathcal{H}$.

Remark: Algorithms to compute the solution of (1) are often based on part (b). See also

1. A. J. Laub, “A Schur method for solving algebraic Riccati equations”, IEEE Trans. Automat. Contr., vol. AC-24, pp, 9l3-921, Dec. 1979
2. The MATLAB documentation of `care`; there you also find a link to
3. Arnold, W.F., III and A.J. Laub, ”Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations,” Proc. IEEE, 72 (1984), pp. 1746-1754

   which is available from `https://engineering.purdue.edu/AAE/Academics/Courses/aae564/2008/fall/Notes/ArnoldLaub1984`

Please turn around!

Task 2:

(a) Give all solutions of the algebraic Riccati equation (1) in the scalar case $n = 1$.

(b) In (a) we saw that the algebraic Riccati equation generalizes the problem of finding the roots of a second order polynomial. Newton's method is a popular method to solve non-linear problems, which is known to be quite successful, when applied to quadratic problems. To use Newton's method to solve (2) we define $\mathcal{R} : H^n \to H^n$ (where H^n denotes the Hermitian matrices of size n-by-n) by

$$\mathcal{R}(X) := X^*GX + F^*X + X^*F - H,$$

and note that $\mathcal{R}(X) = 0$ if and only if X solves (2). To apply Newton's method to this problem consider the following tasks:

1. Compute the (Frechet-)derivative $D\mathcal{R}(X)[\Delta]$.
2. How can one compute the inverse of the linear mapping $D\mathcal{R}(X)$ at some point $\Lambda \in H^n$, i.e., how can one compute $(D\mathcal{R}(X))^{-1}[\Lambda]$. (Hint: Solve a Lyapunov equation)
3. Describe all computations necessary one Newton step.

Task 3:

(a) Let $F : \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^m \to \mathbb{C}^\ell$ and consider a dynamical state-space control system, in which we do not have an explicit formula for $\dot{x}$, i.e., consider a system of the form

$$0 = F(x(t), \dot{x}(t), u(t)), \tag{4}$$

where $x \in \mathcal{C}^n_\infty$ is the state and $u \in \mathcal{C}^m_\infty$ is the input/control. Assume that $\eta_0 := (x_0, 0, u_0) \in \mathbb{C}^{n+n+m}$ is a steady state point, i.e., $F(\eta_0) = F(x_0, 0, u_0) = 0$. Give a linearization of (4) around $(x_0, 0, u_0)$, similar to the last section in lecture, which was called "Non-linear state-space control problems".

(b) Repeat the task for non-linear first-order behavior systems of the form

$$0 = F(z(t), \dot{z}(t)),$$

where $F : \mathbb{C}^q \times \mathbb{C}^q \to \mathbb{C}^p$ and $z \in \mathcal{C}^q_\infty$ around some steady state point $(z_0, 0) \in \mathbb{C}^q \times \mathbb{C}^q$ of F (which means that $F(z_0, 0) = 0$.

(c) Show that (4) is a special case of part (b).

(d) Repeat the task for non-linear higher-order behavior systems of the form

$$0 = F(z(t), \dot{z}(t), \ldots, z^{(K)}(t)),$$

where $F : \underbrace{\mathbb{C}^q \times \ldots \times \mathbb{C}^q}_{K+1 \text{ - times}} \to \mathbb{C}^q$ and $z \in \mathcal{C}^q_\infty$ around some steady state point $\eta_0 := (z_0, 0, \ldots, 0)$.