Application of Control of Convex Monotone Systems

Carolina Lidström and Gustav Nilsson

June 10, 2014

Based on A.Rantzer and B. Bernhardsson. *Control of convex monotone system*. Submitted to CDC 2014.

Outline

Project description

For convex monotone systems, the state trajectory is a convex function of the initial state and the input trajectory

- Want to use the methods from the course to solve an optimal control problem that includes a convex monotone system

Outline

Monotone systems

Definition

A system

$$\dot{x} = f(x, u), \quad x(0) = a$$

is said to be (a controlled) monotone system if its solution satisfies

$$(a_0, u_0) \leq (a_1, u_1) \quad \Rightarrow \Phi_t(a_0, u_0) \leq \Phi_t(a_1, u_1) \quad \forall t$$

Example of monotone systems can be found in: Virus-mutations, Power Networks, Fluid Dynamics...

Monotone systems

Proposition (Rantzer & Bernhardsson 2014, (Angeli & Sontag 2003))

For $f \in C^1$ the following statements are equivalent:

a) *The dynamical system*

$$\dot{x} = f(x, u), \quad x(0) = a$$

is monotone.

b) *The inequalities*

$$\frac{\partial f_i}{\partial x_j} \geq 0, \ \frac{\partial f_i}{\partial u_k} \geq 0, \quad \forall i, j, k \ s.t. \ i \neq j \quad \textit{holds.}$$

c) *The solution to*

$$\dot{x} = f(x(t), u(t)) + v, \quad x(0) = a,$$

is a monotone function of u, v and a.

Convex monotone system

If

$$\dot{x} = f(x, u), \quad x(0) = a$$

is a monotone system and every row of f is convex, the system is called a *convex monotone system*.

Theorem (Rantzer & Bernhardsson 2014)

If $f \in C^1$ and the system is a convex monotone system, then each component of $\Phi_t(a, u)$ is a convex function of a and u.

Convex monotone system

Proof.

$$x_0(t) = \Phi_t(a_0, u_0) \quad x_1(t) = \Phi_t(a_1, u_1)$$

$$\begin{aligned} x_\lambda &= (1-\lambda)x_0 + \lambda x_1 \\ a_\lambda &= (1-\lambda)a_0 + \lambda a_1 \\ u_\lambda &= (1-\lambda)u_0 + \lambda u_1 \\ v &= (1-\lambda)f(x_0, u_0) + \lambda f(x_1, u_1) - f(x_\lambda, u_\lambda) \geq 0 \end{aligned}$$

Let

$$\dot{y}(t) = f(y(t), u_\lambda(t)) + v(t), \quad y(0) = a_\lambda$$

then

$$\phi_t(a_\lambda, u_\lambda) \leq y(t) = x_\lambda(t) = (1-\lambda)\Phi_t(a_0, u_0) + \lambda\Phi_t(a_1, u_1).$$

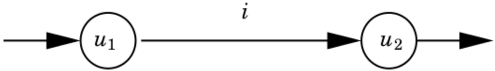
Outline

A dynamical model for power networks - Motivating example

- Two types of nodes, generators and active loads, connected via links (transmission lines)
- An active load tries to keep its power constant by regulating the current

A dynamical model for power networks - Motivating example

Example network: Node 1, u_1, is a generator while node 2, u_2, is an active load.



Dynamical model of current from the active load :

$$\frac{di}{dt} = \frac{\hat{p}}{u_1 - Ri} - i$$

where R is the line resistance.

- The active load tries to keep its power constant at $\hat{p}$ by regulating the current

Dynamical model for power networks - General model

Kirchoff's law for a general network:

$$\begin{bmatrix} -i^G(t) \\ i^L(t) \end{bmatrix} = \begin{bmatrix} Y^{GG} & Y^{GL} \\ Y^{LG} & Y^{LL} \end{bmatrix} \begin{bmatrix} u^G(t) \\ u^L(t) \end{bmatrix}$$

where Y is the admittance (inverse of resistance) and superscript G and L stands for generator and load, respectively. The dynamical model for the active load can then be written as

$$\frac{di_L}{dt}(t) = \hat{p}./[(Y^{LL})^{-1}(i^L - Y^{LG}u^G)] - i^L(t)$$

and for a specific load $k \in 1, .., K$

$$\frac{di_k^L}{dt}(t) = \frac{\hat{p}_k}{u_k^L(t)} - i_k^L(t)$$

The system is convex monotone with state i^L and input $-u^G$.

Convex montone system?

$$f(i, u) = \hat{p}./[(Y^{LL})^{-1}(i^L - Y^{LG}u^G)] - i^L(t)$$

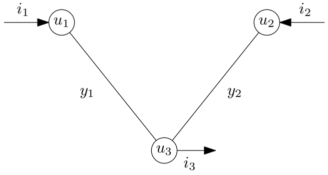
- **Monotonicity** Fact from [Abraham Berman and Robert J. Plemmons. *Nonnegative matrices in the mathematical sciences*]: Structure of $Y^{LL} \Rightarrow (Y^{LL})^{-1} \leq 0$.

  "Idea": $f(x) = \frac{1}{ax+b} \to f'(x) = \frac{1}{(ax+b)^2} \cdot -a$
- **Convexity** $\frac{1}{g(x)} + h(x)$ where $g(x) > 0$ and $h(x)$ are affine.

Outline

Numerical example

Network: Two generators, u_1 and u_2, and one active load, u_3



Kirchoff's law:

$$\begin{bmatrix} -i_1(t) \\ -i_2(t) \\ i_3(t) \end{bmatrix} = \begin{bmatrix} -y_1 & 0 & y_1 \\ 0 & -y_2 & y_2 \\ y_1 & y_2 & -y_1 - y_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix}$$

Dynamics:

$$\frac{di_3}{dt}(t) = \frac{\hat{p}}{u_3(t)} - i_3(t) = \frac{\hat{p}(y_1 + y_2)}{y_1 u_1(t) + y_2 u_2(t) - i_3(t)} - i_3(t)$$

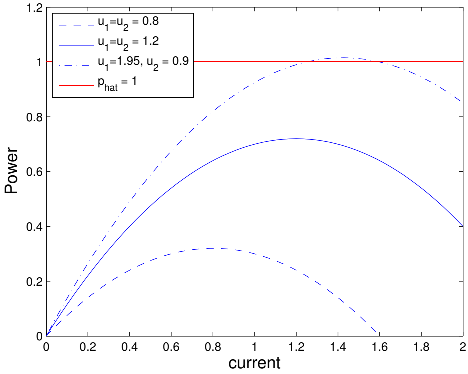
Voltage collapse

Dynamics:

$$\frac{di_3}{dt} = \frac{\hat{p}(y_1 + y_2)}{y_1 u_1 + y_2 u_2 - i_3} - i_3$$

Power in node 3:

$$p_3(t) = u_3(t) \cdot i_3(t)$$



Outline

The optimal control problem

Input: u_1 and u_2

$$
\begin{aligned}
\text{minimize } & t_f^2 \\
\text{subject to } & i_3(t_f) \cdot u_3(t_f) = \hat{p} \\
& \frac{di_3}{dt}(t) = \frac{\hat{p}}{u_3(t)} - i_3(t) \\
& \begin{bmatrix} -i_1(t) \\ -i_2(t) \\ i_3(t) \end{bmatrix} = \begin{bmatrix} -y_1 & 0 & y_1 \\ 0 & -y_2 & y_2 \\ y_1 & y_2 & -y_1 - y_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} \\
& \dot{u}_1 \leq 1 \quad \dot{u}_2 \leq 0.5
\end{aligned}
$$

Implementation of Optimal control problem in Jmodelica

```
model PowerNetwork

input Real u1p;
input Real u2p;

Real u1(start =1.3, fixed = true);
Real u2(start = 0.6, fixed = true);
Real i3(start = 1.2, fixed = false);

Real p;
Real u3;
Real p3;
Real i2;
Real i1;

constant Real y1 = 1;
constant Real y2 = 1;
constant Real p3_hat = 1;
equation
// Integrator
der(u1) = u1p;
der(u2) = u2p;
// Active load
der(i3) = p3_hat * (y1 + y2) / (y1 * u1 + y2 * u2 - i3) - i3;

// Currents Kirchhoff law
0 = (-i1) - i2 + i3;
0 = i1 - y1 * u1 + y1 * u3;
0 = i2 - y2 * u2 + y2 * u3;

p =(-i1) - i2 + i3;
p3 = i3 * u3;
end PowerNetwork;
```

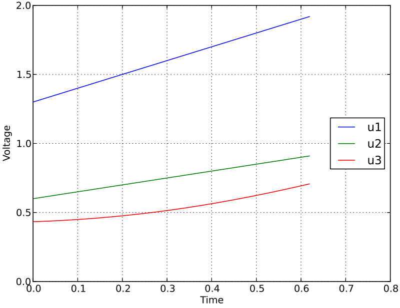
Implementation of Optimal control problem in Jmodelica

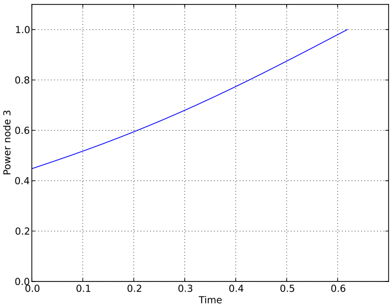
```
optimization PowerNetwork_MinTime(finalTime(free=true, min=startTime),
                                  objective=(finalTime^2))
    extends PowerNetwork(u1(min=0.05,max=10), u2(min=0.05,max=10));

constraint
u1p <= 1;
u2p <= 0.5;
p3(finalTime) = p3_hat;
end PowerNetwork_MinTime;
```

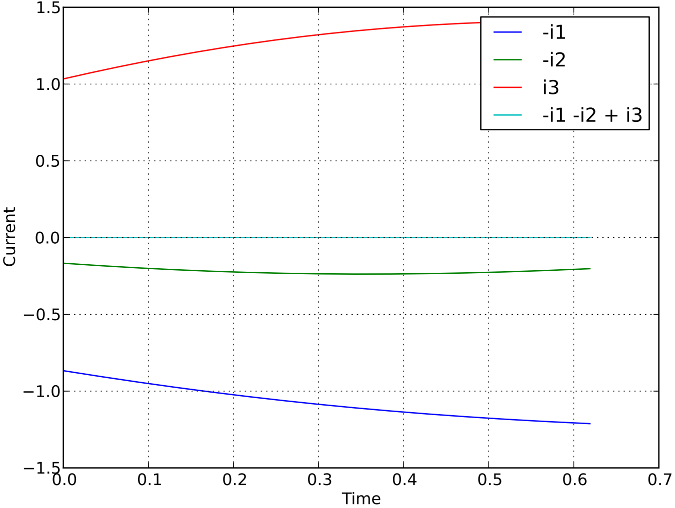
Outline

Result


Voltage
Time
u1
u2
u3
2.0
1.5
1.0
0.5
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Power node 3
Time
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Result



-i1
-i2
i3
-i1 -i2 + i3
Current
Time
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

The optimal control problem - fixed final time

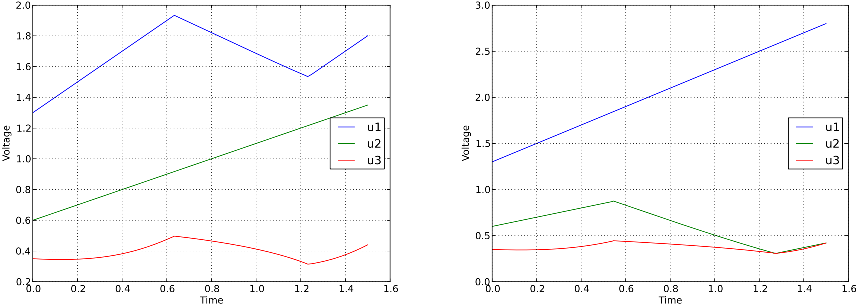
Input: u_1 and u_2

$$\begin{aligned}
\text{minimize} \quad & \int_0^{1.5} 10 \cdot u_1(t)^2 + u_2(t)^2 dt \\
\text{verusus minimize} \quad & \int_0^{1.5} u_1(t)^2 + 10 \cdot u_2(t)^2 dt \\
\text{subject to} \quad & i_3(t_f) \cdot u_3(t_f) = \hat{p} \\
& \frac{di_3}{dt}(t) = \frac{\hat{p}}{u_3(t)} - i_3(t) \\
& \begin{bmatrix} -i_1(t) \\ -i_2(t) \\ i_3(t) \end{bmatrix} = \begin{bmatrix} -y_1 & 0 & y_1 \\ 0 & -y_2 & y_2 \\ y_1 & y_2 & -y_1 - y_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix}
\end{aligned}$$

The optimal control problem - fixed final time

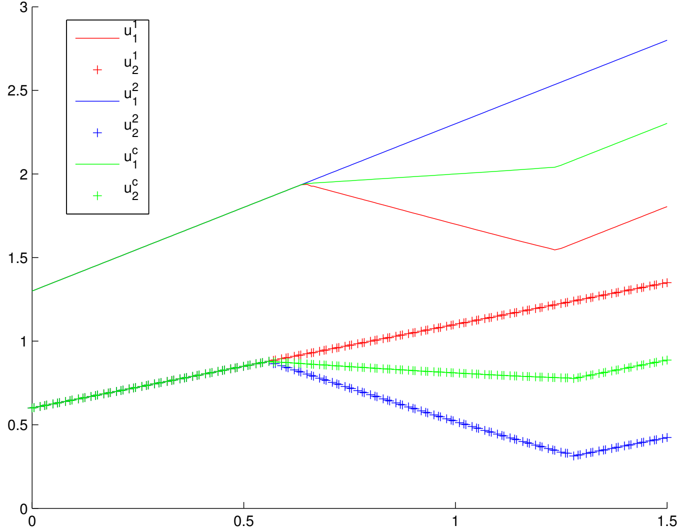
```
 optimization PowerNetwork_Lagrange1(finalTime=1.5, objectiveIntegrand=10*u1^2+u2^2)
    extends PowerNetwork(u1(min=0.05,max=10), u2(min=0.05,max=10));
constraint
i1 >= 0;
i2 >= 0;
u1p <= 1;
u2p <= 0.5;
p3(finalTime) = p3_hat;
end PowerNetwork_Lagrange1;
```

Simulation results



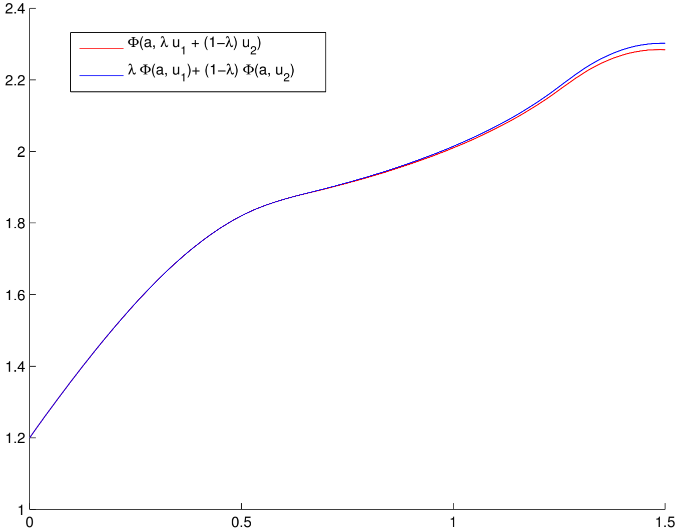
Convex combination of control signal

$\lambda = 0.5$



Convexity of the system

$\lambda = 0.5$



Outline

Comments on results

- Time-optimal control of monotone systems, "maximize" the control signal
- Convex-monotone system - still open question how to use the convexity property?