Function Electrically Stimulation of the Calf Muscle Using Matlab and Polynomial Toolbox

Petr Urban[*,**], *Michael Šebek*[*,**]

[*]Department of Control Engineering, Faculty of Electrical Engineering,
Czech Technical University, Prague
[**]Institute of Information Theory and Automation,
Czech Academy of Sciences, Prague

Abstract

This work is concerned with the design of robust control systems for the control of ankle moments generated by the electrical stimulation of the human calf muscle. We try to design robust controllers using functions of Polynomial Toolbox 3.0 and SeDuMi. This work is made in cooperation with Centre for Rehabilitation Engineering in University of Glasgow.

1 Introduction

This work is engaged in robust control of electrically-stimulated muscle of people with spinal cord injuries. In this paper we present a simple techniques to design the robust controller of force (moment) generated at human ankle joint by electrically stimulated calf muscles. This is a part of the problem of designing controller that stabilize the vertical posture of paraplegia when they would like stand (unsupported standing).

This work is made in cooperation with Centre for Rehabilitation Engineering in University of Glasgow. Our work on this problem is design controllers that could have been used in practical application. The main idea of this work is to create small box that includes controller and other necessary tools for unsupported standing, walking, cycling etc. Other work like identification of the transfer functions of the muscles and practical experiments are doing by people around Prof. Ken Hunt at the Department of Mechanical Engineering, University of Glasgow. He is header of Centre for Rehabilitation Engineering - CRE, for more details about this group you can see [2].

2 The System description

People from CRE have been developing feedback systems for control of unsupported standing in complete paraplegia (namely Donaldson, 1993; Donaldson, Barr, Phillips & Perkins, 1996; Hunt, Munih, Donaldson & Barr, 1998b; Hunt et al., 1997; Munih et al., 1997). In its simplest form the body is regarded as a single-link inverted pendulum with moment only around the ankle joint, see Figure 1. Where $\theta(t)$ is the inclination angle moment and $p(t)$ is the pulse width of the stimulation. C_m is the moment controller and C_θ is the angle controller. The desired values for ankle moment and inclination angle are m_{ref} and θ_{ref}, respectively.

Table 1: Parameters of the identified plant models

Model	Transfer function	Bandwidth (rad/s)	DC-gain (Nm/μs)
1	$\frac{0{,}02708q^{-1}}{1-0.9662q^{-1}+0.254q^{-2}}$	11.5	0.094
2	$\frac{0{,}03311q^{-1}}{1-1.168q^{-1}+0.3766q^{-2}}$	9.9	0.1587
3	$\frac{0{,}02671q^{-1}}{1-1.248q^{-1}+0.3882q^{-2}}$	6.1	0.1905
4	$\frac{0{,}01885q^{-1}}{1-1.174q^{-1}+0.2641q^{-2}}$	2.8	0.2092

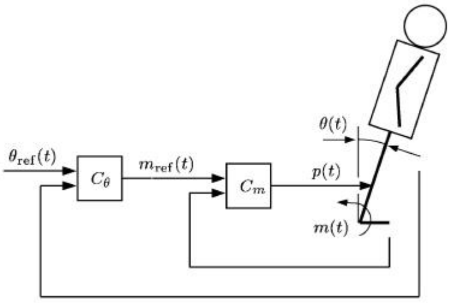

Figure 1: Control loop

Experimentally, this situation has been realized by standing subject in an apparatus known as the "Wobbler" (Donaldson, Munih, Philips & Perkins, 1997), as shown in Figure 2. For more information you can see [1] or [2].

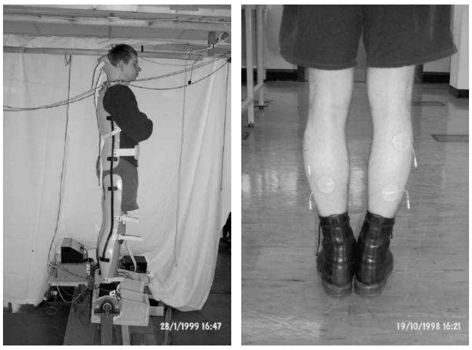

Figure 2: Subject standing in Wobbler and placement of electrodes on the calf muscle.

Transfer functions of models are listed in Table 1 and their characteristics are shown in Figure 3. They were obtained from ARX model identification. This problem is described in [1]. Against the [1] we have reduce number of models because one of models haven't correspond with presented data and characteristic.

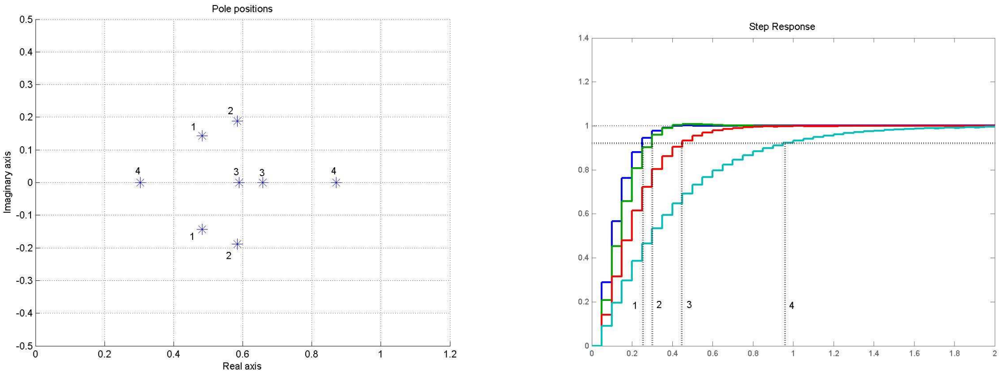

Figure 3: Characteristic of the local models identified in 4 points.

3 Design of controller

For design of the controller we have used POLYNOMIAL TOOLBOX 3.0, see [4]. In this last version functions from Didier Henrion are implemented. These functions use LMI methods implemented in the LMI solver SEDUMI [5] and SEDUMI INTERFACE [6]. For more details about these functions you can see [3].

How we can design the robust controller using these functions:
Let the $P_i = N_i/D_i$, i $= 1, ..., 4$ is the set of transfer functions all models.

1. Like a first we must select the nominal model. Our choice is Model 3, because it was identified in the mid-range operating point.

2. Using the function *ptopdes* we obtain only controller that stabilize our models but we need provide the zero steady-state error. So we must add to the controller the integrating part $(1-q)^{-1}$. In praxis this means that we add the integrating part to models transfer functions.
$$P_i = \frac{N_i}{D_i}\frac{q}{(q-1)} = \frac{N_i\_new}{D_i\_new}$$

3. Now we must design a controller that stabilize the nominal model. For this we use the function *pplace* (for help you can see [4]):

```
poles=[0.3,-0.3,0.3-0.7j,0.3+0.7j,0.1-0.2j,0.1+0.2j];
[n,d]=pplace(N_new{jj},D_new{jj},poles);
```

4. Using
$$Y * N\_new(3) + X * D\_new(3) = CL$$
we obtain the central (nominal) closed loop polynomial. Now we can choose the stability region using the matrix S. Default setting for discrete time polynomials is the unit circle
$$\mathbf{S} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$
but our choice is the circle with the radius 0.8 and then
$$\mathbf{S} = \begin{bmatrix} -0.8^2 & 0 \\ 0 & 1 \end{bmatrix},$$

For more details about choosing matrix **S** you can see [3].

5. The next step is design of the robust controller using *ptopdes*:

```
[X,Y]=ptopdes(N_new,D_new,CL,S);
```

   where N_{new} and D_{new} are cell arrays:

```
N_new{1} = 0.02708*q^1 * q;
D_new{1} = 1*q^2 - 0.9662*q^1 + 0.254) * (q-1);

N_new{2} = 0.03311*q^1 * q;
D_new{2} = 1*q^2 - 1.168*q^1 + 0.3766) * (q-1);

N_new{3} = 0.02671*q^1 * q;
D_new{3} = 1*q^2 - 1.248*q^1 + 0.3882) * (q-1);

N_new{4} = 0.01885*q^1 * q;
D_new{4} = 1*q^2 - 1.174*q^1 + 0.2641) * (q-1);
```

6. Last step consists from adding the integrating part back to the controller:

$$C = \frac{X}{Y}\frac{(q-1)}{q}$$

4 Simulations

In this section the results of controller design are presented. Simulations have been made in MATLAB R14 and MATLAB SIMULINK. The designed controller has this transfer function:

$$C = \frac{0.02192z^4 - 0.005713z^3 - 0.007204z^2 - 0.008468z - 0.0005313}{z^4 - 0.4699z^3 + 0.1536z^2 - 0.2932z}$$

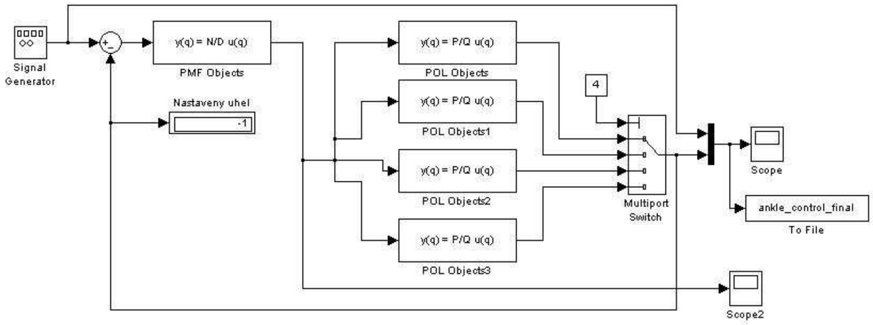

Figure 4: Simulink schema of the feedback moment control.

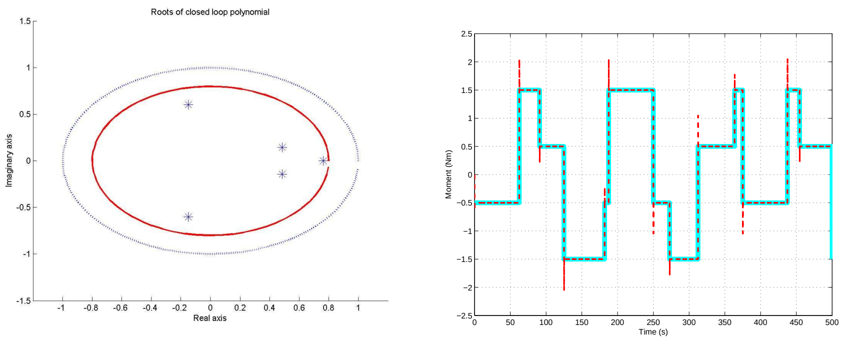

Figure 5: Roots of closed loop polynomial and result of the simulation - the bold line is the reference, the dashed line is the ankle moment.

5 Conclusions

Based on results it seem to be the LMI polynomial methods good candidate for design of controllers for this problem, see Figure 5. We show only one example of the result of simulation because all simulation has the same result for all four models. But still we must be careful because these results are only from theoretical simulations. In practical experiment we must assume that there are much differences e.g. between a healthy person and a paraplegia (the healthy person has different sensitivity threshold and well-developed muscles than the paraplegia). So differences between models are much more different. Here is the problem of models identification and control constraints.

6 Acknowledgements

This work was partly supported by the grant 102/03/H116 of the Grant Agency of the Czech Republic.

References

[1] Hunt K.J., Jaime R.P., Golle H.: Robust control of electrically-stimulated muscle using polynomial H_∞ design. Control Engineering Practice 9, p.313-328, (2001).

[2] Ken Hunt: Centre for Rehabilitation Engineering:<http://fesnet.eng.gla.ac.uk/CRE/>.

[3] Henrion, D., Šebek, M.: New robust control functions for the Polynomial Toolbox 3.0, 2002, May 17.

[4] PolyX, Ltd.: Polynomial Toolbox 3.0, (prelease):<http://www.polyx.com>.

[5] Jos F. Sturm: SeDuMi:<http://fewcal.kub.nl/sturm/software/sedumi.html>.

[6] Peaucelle, D.,Taitz, K.: SeDuMi Interface:<http://www.laas.fr/~peaucell/SeDuMiInt.html>.