

Attempting sliding mode controller to mobile robot arc welding process.

Zbigniew Mrozek, Stanislaw.Tarasiewicz
Politechnika Krakowska, Poland; Université Laval, Québec, Canada
zbigniew.mrozek@pk.edu.pl stanislaw.tarasiewicz@gmc.ulaval.ca

Abstract: *The Mobile Robot Arc Welding (MRAC) process is very complex one. For this application, the mobile platform and its feed driving motor will be presented. Using MATLAB/SIMULINK software it was possible to simulate the automatic control process of the mobile platform movement. STATEFLOW was used to design (reactive) controller behaviour using finite state machine diagrams.*

1. Description of The Mobile Robot Arc Welding (MRAC) process. The MRAC process is very complex one. This complexity is caused by wide variation of mechanical and physical properties integrated with the mobile platform. Several design problems such as mechanical structure of the platform and automatic control of the mobile system should be solved. Depending on those parameters, the MRAC has been divided into two main subsystems:

1. *IRB2400L* robot from *ABB* company.
2. its platform mobile with feed driving motor

For this application, the mobile platform and its feed driving motor will be presented.

1.1 Drive of the platform mobile. Chassis and DC drive of *Robomac100* (figure 1) was used to build platform mobile (PFM) for arc welding system. PFM was adopted for movement on rails. DC motor (3HP, 1750 rpm) is fed with variable speed full-wave regenerative drive capable of operating DC motors in a bi-directional mode. Its 4-quadrant operation provides forward and reverse torque in both speed directions. This allows the control to maintain constant speed with overhauling loads and provides rapid instant reversing and

controlled braking. Dodge reducer with extra taper-lock sprocket is used. Now 100 revolutions of DC motor shaft results 1526 mm displacement. PLC is used as control hardware.

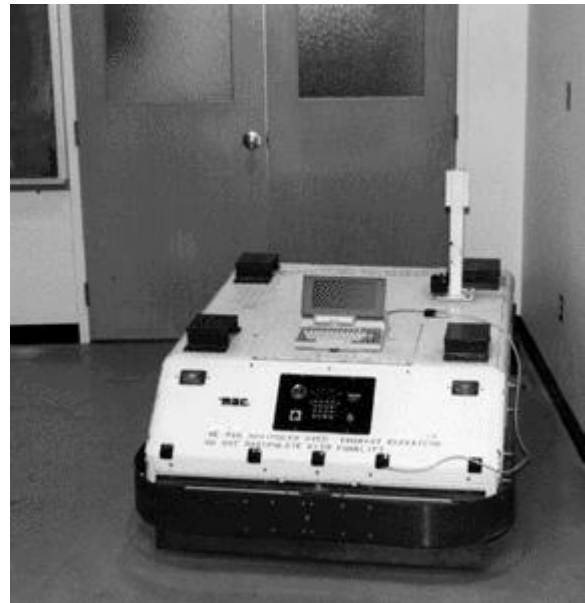


Fig. 1 *Robomac100* was used to build platform mobile for arc welding system

2. Sliding mode controller. Using the sliding approach it is possible to minimize the positioning errors. This approach gives also fast response, good transient performance and robustness with regard to parameter variations and external disturbances [3].

2.1 An example of sliding mode controller. The control goal is to move the object from any initial condition (IC) to the origin (0,0).

Second order model with damped oscillations is used as an example object :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\beta\omega \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u \quad (1)$$

Attempting sliding mode controller will be described using *MATLAB* linear time invariant (LTI) model :

```
omeg=2*pi*2;
bet=.2;
A=[0 1; -omeg^2 -2*bet*omeg];
B=[0, omeq^2]';
C=[1 0; 0 1]; D=0;
sysTST=ss(A,B,C,D);
```

Step response $x_1(t)$, its derivative $x_2(t) = dx_1/dt$ and phase portrait $x_2 = f(x_1)$ of system (1) is presented on figure 2.

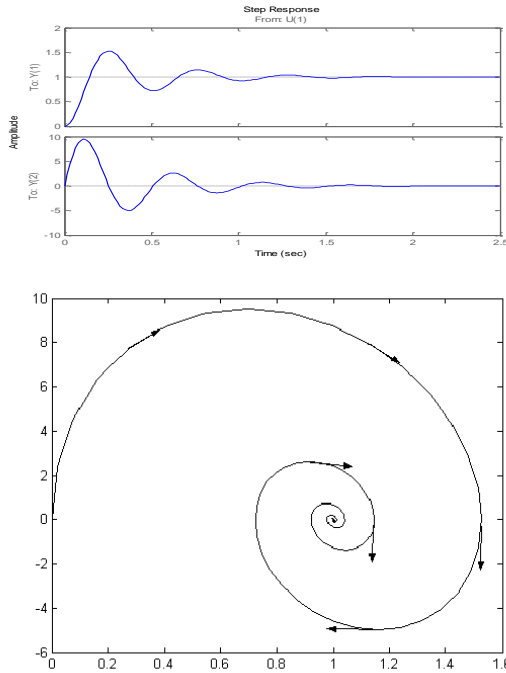


Fig. 2. Step response and phase portrait of system (1)

The model properties will be changed with state feedback. The feedback gain is found using *root locus* technique. Alternatively *MATLAB* function *poles* or *SISO design tool* can be used. The gains are :

- $kH=[-1.9119, 0.0125]$ for unstable saddle type aperiodic response
- $kO=[-0.2084, -0.0952]$ for unstable oscillatory response

Phase portrait for closed loop feedback system is presented on figure 3. Initial condition starting points are marked with a small circle.

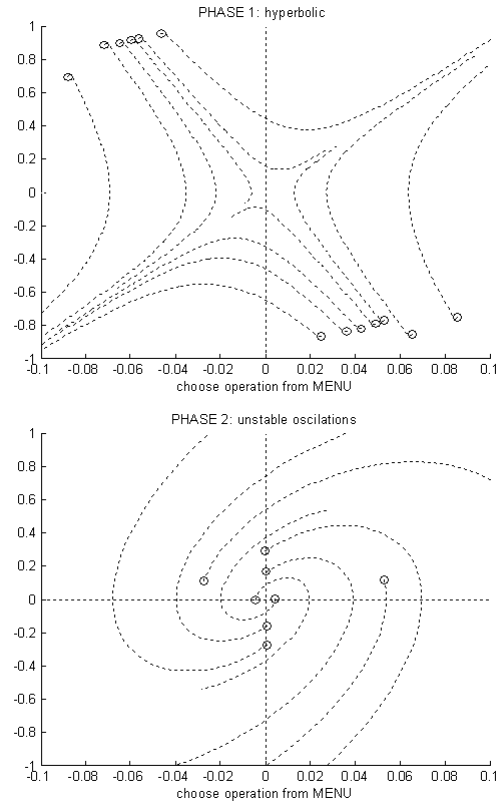


Fig. 3 Phase portrait for closed loop feedback system

New system properties will be obtained with sliding mode controller. A desired stable trajectory will be build using portions nonstable aperiodic and oscillatory responses of the system. A switching line s (or switching surface in multi-dimensional phase plane) can be defined. Starting from any initial condition, only trajectories leading to the switching line will be considered (lower part of figure 4). When switching line is achieved and passed, the control law should change structure of the system. The goal is to change direction of the movement. So another trajectory is used, leading back to switching line from the other side. Now switching is very fast and the system is kept in

neighborhood of the switching line. This new trajectory does not depend on properties of the original system. This behaviour is called a sliding mode.

1.2 Implementation of sliding mode controller

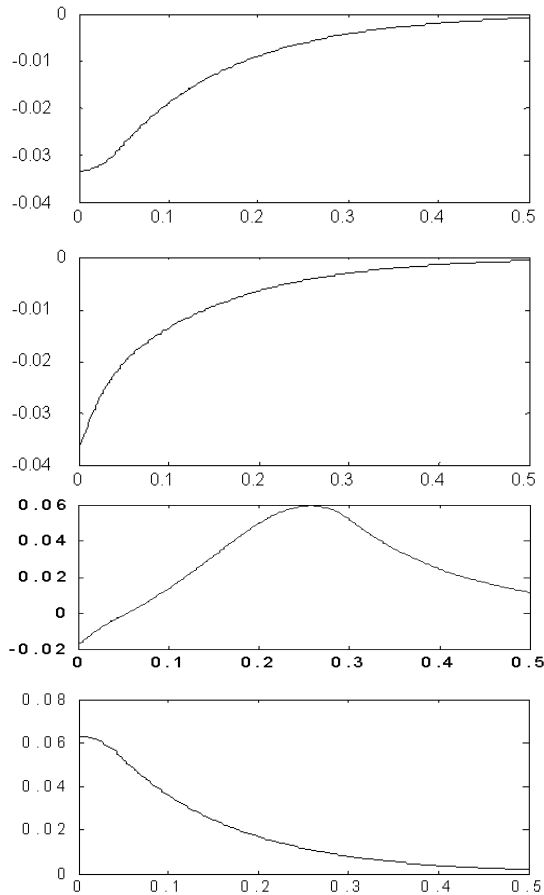


Fig. 4 Time response phase portrait of the system with sliding mode controller. Initial condition starting points are marked with A,B,C,D letters.

MATLAB, *SIMULINK* and *STATEFLOW* were used to prepare environment to design (reactive) controller behaviour. Graphical user interface is build to help designing sliding mode controller using *STATEFLOW* finite state machine diagrams. Original system equation is given in M-file. Poles location for non-stable aperiodic and oscillating operation are chosen by user. Then **place** function from *MATLAB Control Toolbox* is used to find feedback gain vectors. At last a new graphical window (figure 5) is opened.

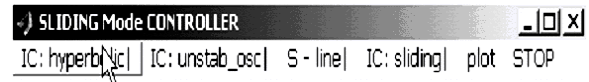


Fig. 5 Menu of sliding mode design window

Using the menu one can draw family of trajectories (see fig.3). Then position of switching line is chosen and one may observe sliding mode operation (see bottom of figure 4) starting from any initial condition shown on the computer screen with mouse. Vertical axis x_2 may serve as auxiliary switching line. This is needed to prevent running to infinity from some initial condition (e.g. from point C on figure 4). Conditions to guarantee that sliding line is reachable from any initial condition can be found in [3].

The control law of sliding mode controller is programmed graphically with *STATEFLOW* state diagram [2] presented on figure 6.

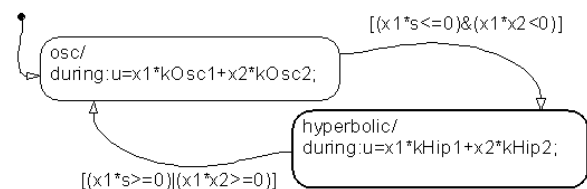


Fig. 6. State diagram for sliding mode controller

The system will change a structure when condition given on transition line is satisfied.

3 Identification of PFM parameters.. The square wave velocity input signal was generated with *Allen Bradley Micrologix 1500* PLC controller. The input signal was chosen to be : 0.2, 0.3 ... 1.0, where 1.0 is nominal

velocity of the PFM system. The PFM system output was actual distance from starting position. This was measured with encoder mounted on driving axle. The experiment data was registered on PC computer as *Excel* file. RS232 link was used for communication between PC computer and PLC controller. *System Identification Toolbox* of *MATLAB* was used to find the relationship between the mechanical structure of this platform and the motor input. Two data sets were used: first for identification, another for verification.

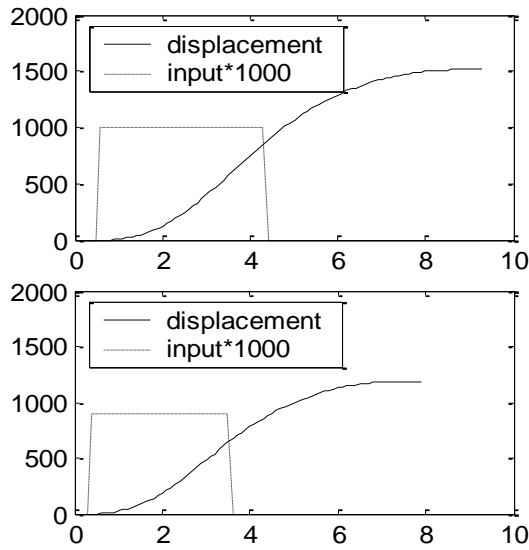


Fig. 7. Experimental data of PFM movement

The system response is aperiodic (figure 7) and low order model **arx211** shows good agreement with experimental data (figure 8).

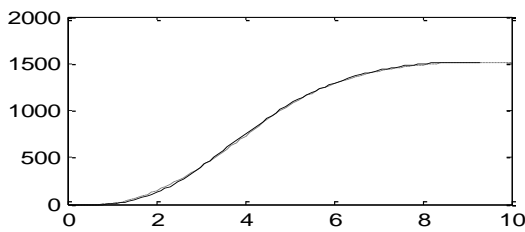


Fig. 8. Comparison of experimental (dotted line) and simulation results (solid line)

The discrete model for PFM object was :

```
>> tf211
      Transfer function:
          1.878
      -----
      z^2 - 1.963 z + 0.9634
      Sampling time: 0.1
```

3. Sliding mode controller

Continuous state space equation of PFM system was found using **d2c** and **ss** functions from *MATLAB Control Toolbox*. The system matrix is:

$$A = \begin{bmatrix} 9.8141 & 4.9075 \\ 20.3759 & 10.1870 \end{bmatrix}$$

System output is given by $y=Cx$ equation and dy/dt is calculated numerically. The **place** function is used to find state feedback vector for nonstable oscillatory and saddle type response of the PFM system. Simulation results are given below on figure 9 and 10.

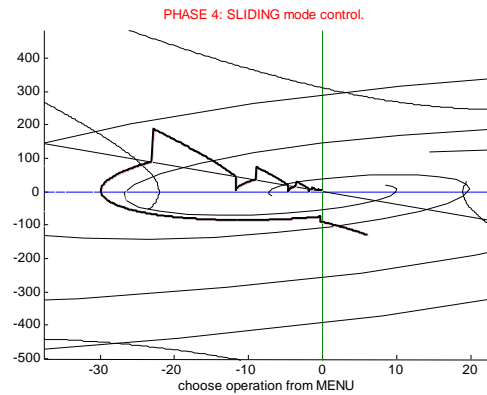


Fig. 9 Phase portrait for PFM with sliding mode control. High IC derivative component.

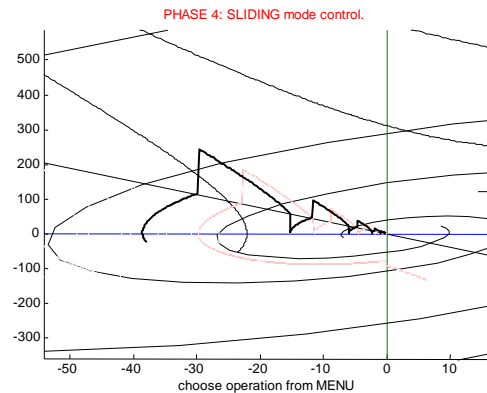


Fig. 10 Phase portrait and time response for PFM with sliding mode control

Figure 10 shows the phase portrait for PFM with sliding mode control. High derivative component is included in initial condition. This example shows that switching line can be reached from different regions of phase plane.

More practical example is shown on figure 11. Actual IC has small derivative component. The output has no overshoot and settling time of the time response is shorter than above.

Designing sliding mode controller for existing PFM object was more difficult then described in example in paragraph 2.1. The state variables of PFM system are not directly measurable. Derivative of output signal (displacement) was not a state variable as before. It was observed that switching the controller structure instantly changes value of the output signal derivative (compare figures 9 and 10 with figure 4).

Time lag of digital version of sliding mode controller (not described here) was too high comparing with non-stable system response and switching the controller structure did not work properly even for discretisation time equal 2 ms.

4. Designing discrete PID controller for PFM. Sliding mode controller operation will be compared with discrete PID controller. The PID coefficients: $K_p=0.7488$, $K_i=3.2918e-004$ and $K_d=1.0606e-004$ were found using *SIMILINK* and *NCD (Nonlinear Control Design) Toolkit*.

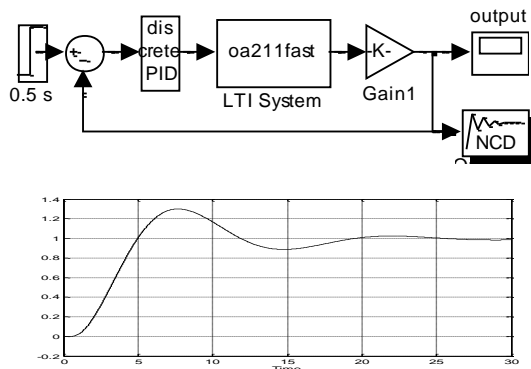


Fig. 11 Designing PID controller for PFM with NCD Toolkit. Step response of the system

Transient response of PFM object with PID controller seems to have too long settling time

and too big overshoot. The results are not as good as for sliding mode controller.

Conclusion

The paper describes methodology and simulation results for attempting sliding mode controller for PFM movement control. Future experimental validation of simulation results will be essential for the described problem. Using encoder output (and its derivative) as a feedback signal should be preferred comparing with a state feedback, as presented here. The problem is to obtain non-stable oscillatory and saddle type response of the closed loop system without state feedback.

Literature

1. Bühler H, Réglage par mode de glissement, Presses Polytechniques Romandes, Lausanne 1986,
2. Mrozek B, Mrozek Z. MATLAB 6, poradnik użytkownika, PLJ Warszawa, 2001.
3. Utkin V.I, Variable structure systems with sliding modes, IEEE Trans. AC-22(2) pp212-222
4. Utkin V.I et al, Sliding Control in Electromechanical Systems (book), 1999