AVR191: Anti Pinch Algorithm for AVR Adaptation Procedure

Automotive Microcontrollers

Application Note

Introduction

The purpose of this document is to explain how to adapt an anti-pinch algorithm to a specified powered window. This algorithm is described in AVR480 application note which can be found on Atmel website. Algorithm is adaptive, but some parameters have to be manually set. This application note describes how motor, mechanical and electrical parameters have to be set to allow correct function of the anti-pinch algorithm descirbed in AVR480.

Algorithm overview

Anti-pinching systems are specified by standards which describe pinching effort and pinching detection areas. Other constraints are taken into account, as described in anti-pinch application note AVR480.

Anti-pinch algoritm is able to adapt itself to various friction values thanks to current measurments and derivative speed computation. Both values lead to determine a current reference.

Robustness has been included in algorithm through current average and blocking point detection.

Pinching conditions are detected using a comparison between current reference and pinch threshold composed of a constant and of a permanently computed blocking point information. Porting algorithm from a window lift to another requires to adapt this pinch threshold. This requires:

- Mechnical paremeters knowledge.
- DC motor parameters knowledge.
- Acquisition chain knowledge.
- Standards

Configuration

All parameters allow to configure the algorithm. It can be considered as a database, updatable to several opening aparatus.

They can be set up in non-volatile memroy (using MCU EEPROM) as well as in source code before compilation (using a definition file).

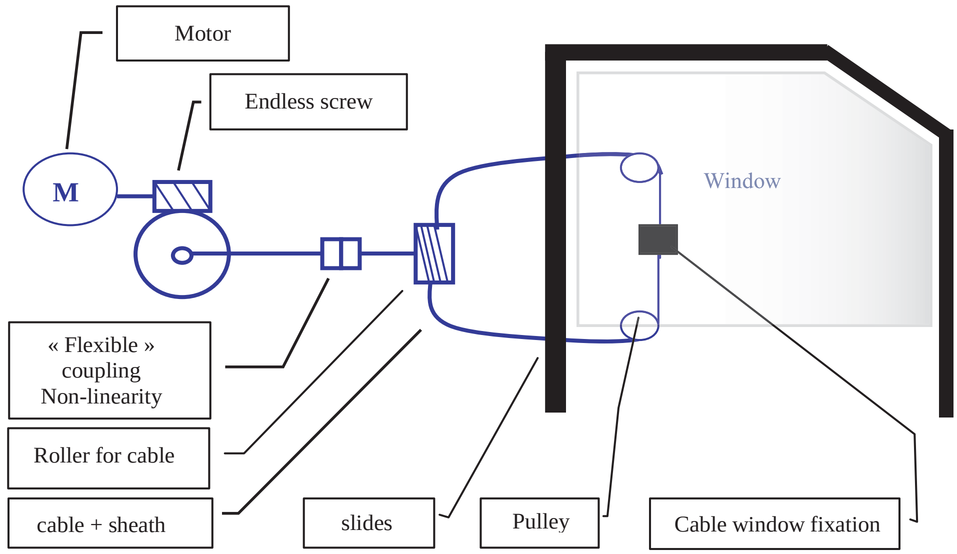
7678A–AUTO–11/06

Required parameters

Anti-pinch application note AVR480 describes a window lift model. This section describes which parameters shall be updated when porting the algorithm.

Mechanical parameters

Figure 1. Powered Window Mechanical Components



For the algorithm to work on the window lift, the following parameters have to be known:

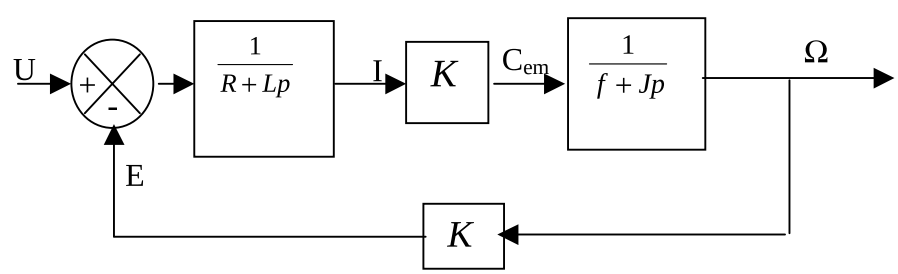
- **r**: The roller cable radius
- **n**: The reducer ratio
- **Window height**: the movement magnitude of the window moving part
- **Rubber joint deep**: (In order to inhibit anti-pinch into the upper part of the rubber joint)

The anti-pinch algorithm is adaptive to various frictions and the adaptation routine (executed to initialize the window-lift) will identify and store blocking points and non-linearity magnitude.

DC Motor parameters

A second order motor model has been chosen to implement the anti-pinch algorithm.

Figure 2. Second order motor model

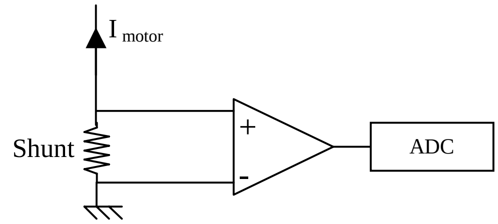


You need to know or to determine motor torque constant **K**. Other parameters may also be useful to determine it, but are not required by algorithm (See appendix 5.2 “Motor Identification example”).

Acquisition chain

The algorithm requires motor current acquisition. Acquisition chain has to be identified.

Figure 3. Global Acquisition chain (from current value to ADC register value)



The global acquisition chain gain has to be determined to obtain:

$$G_{acq} = \frac{ADC}{I_{motor}} \Rightarrow ADC = G_{acq} \times I_{motor}$$

Parameters from standards

Following parameters are extracted from standards:

- **Pinch threshold**: In addition, the user will have to parameter the pinch threshold value in Newton. This value shall be less than 100N according to standards.
- **Higher position**: Correspond to rubber joints deep + 4mm (All positions are referenced from the top of the window lift). Pinch conditions are not checked above this position
- **Lower position:** Pinch conditions are not checked under this position (typical is 200mm from the top of the window)

Parameters summary

- **r**: The roller cable radius
- **n**: The reducer ratio
- **Window height**: the movement magnitude of the window moving part.
- **Rubber joint deep**: (In order to inhibit anti-pinch into rubber joint)
- **K**: Motor torque constant
- **Gain Acq**: from Acquisition chain
- **Pinch threshold**
- **Higher position**
- **Lower position**

Computing Parameters

Gain

The algorithm computes a current reference (multilied by $Gain_{Acq}$ factor). To detect a pinch, it compares: ADC acquired value to reference + Pinch value.

Pinch value has to be in range with ADC values. ADC represents a measured current. Pinch has to represent also a current value. So, we have:

$$\begin{cases} Cem = K \times I \\ C_{pinch} = \frac{r}{n} \times F_{pinch} \end{cases} \Rightarrow I_{pinch} = \frac{r}{n \times K} \times F_{pinch}$$

where F_{pinch} is the pinch threshold value.

Using acquisition gain,

$$ADC = I_{motor} \times Gain_{Acq}$$

The global gain is defined:

$$\left.\begin{array}{l} Pinch = I_{pinch} \times G_{acq} \\ Pinch = \frac{r \times G_{acq}}{n \times K} \times F_{pinch} \end{array}\right\} \Rightarrow Gain = \frac{r \times G_{acq}}{n \times K}$$

Pinch value is in range with ADC values and directly represents the force applied to the window lift thanks to **Gain** value, provided by user.

Other parameters

Number of hall sensors used shall be defined. Indeed, operating direction can be measured when using two hall sensors, which provides more positionning precision. When only one hall sensor is used, command direction is used to compute position.

Parameters from standards:

Pinch threshold: It could be determined by statistic methods using repetitive pinch measurements. It must provide a satisfying probability not to have pinches greater than 100N.

Upper glass position and **Lower position** to detect a pinch: Those values shall be filled in millimeters. Algorithm will convert it then, depending on hall position sensor and multi-pole magnetic ring resolution, thanks to adaptation routine. (All positions are referenced from the top of the window lift).

Some parameters are automatically filled by adaptation routine. Those could be found in Appendix “5.1 Anti-pinch Parameters storage”

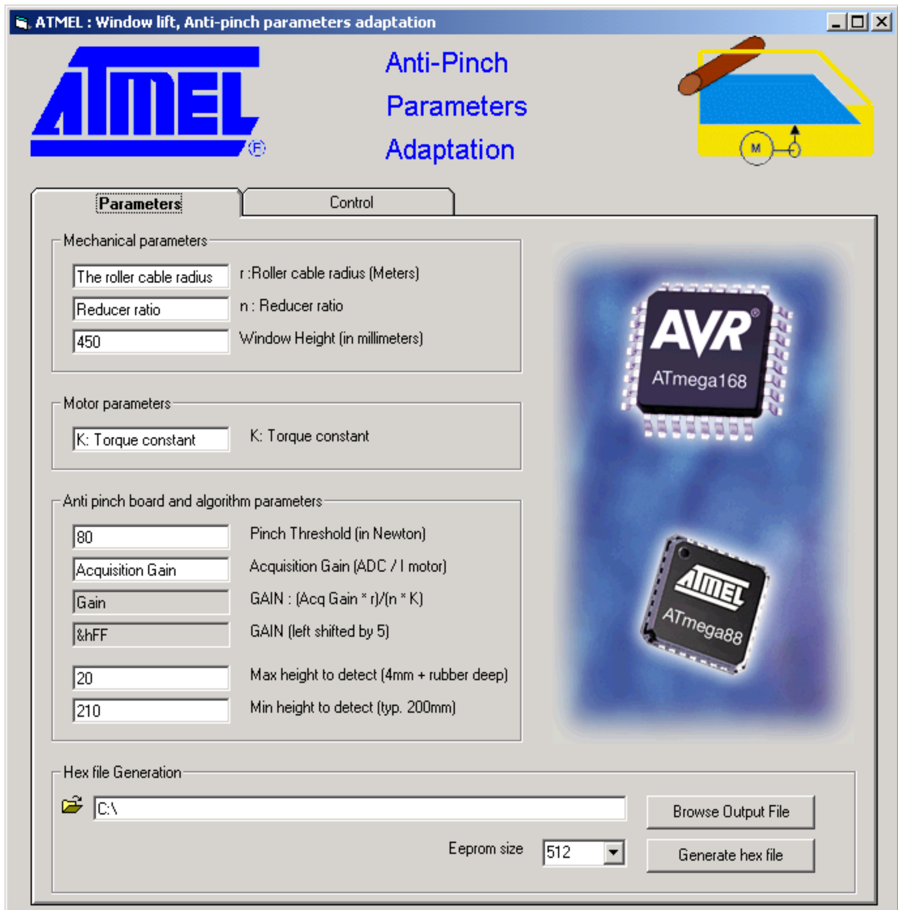
Setup parameters

Using HMI

Previously determined parameters can be stored to EEPROM, to be used as a database by anti-pinch algorithm. It has to be stored with the right format, as described in appendix “5.1 Anti-pinch Parameters storage”.

For this, executing HMI is very useful. Parameters entered are computed and can be correctly formatted. A hexa file is generated and is directly storable to EEPROM (using AVR studio).

Figure 4. Anti-pinch parameters adaptation HMI



Source code setup

All those parameters can be set up in source code as default values. They are all defined in window_lib.h. Those will be used in case of EEPROM has never been initialized and is filled with 0xFF.

The Number of hall effect sensors shall also be defined in config.h configuration file.

Parameters Updated by adaptation routine

In addition to blocking points detection, some parameters are acquired at initialization time by adaptation routine:

- Hall effect sensor resolution
- Top and bottom positions
- Backlash

A first complete downward, then upward operation allow to count the number of hall effect sensor pulses for a complete operation. It is stored in EEPROM as “down” value (see Appendix “5.1 Anti-pinch Parameters storage”). It’s used to determine the resolution and to calibrate the anti-pinch algorithm working areas.

In a second time, adaptation routine operates window-lift downward, into pinch detection area (between 4mm and 20mm from top). It reverses direction upward and monitors pinch condition to measure backlash (used in startup phase of the window lift state machine (see application note). Indeed, start current is seen like a pinch: adaptation routine stores the number of hall sensor pulses to inhibit anti-pinch at motor startup (see Appendix “5.1 Anti-pinch Parameters storage”).

Appendix

Anti-pinch Parameters storage

Anti-pinch parameters are stored, retrieved and refreshed in a table.

Table 1. Stored parameters to non volatile memory (eeprom)

Description	Offset	Comments	Updated by adaptation routine
Position lsb	0	Window-lift position	Yes (and by application)
Position msb	1		
Last Direction	2	Last operating direction	Yes (and by application)
Pinch threshold	3	Pinch threshold value	No, Filled by user
Gain	4	see §“3.1 Gain”	No, Filled by user
Window height lsb	5	Window height (mm)	No, Filled by user
Window height msb	6		
Higher position	7	Distance from top (to inactivate anti-pinch)	No, Filled by user
Lower position lsb	8	Distance from top (to activate anti pinch)	No, Filled by user
Lower position msb	9		
Down lsb	10	Distance from top to bottom (number of measured hall sensor pulses)	Yes
Down msb	11		
Backlash lsb	12	Measured non-linearity when starting up operation, as last operation was a downward	Yes
Backlash msb	13		
-	-	-	-
Blocking point nbr	15	Registred blocking point number	Yes, and by application
Blocking points	16	1st registered blocking point	Yes, and by application
“	20		
“	21	2nd registered blocking point	Yes, and by application
“	25		
“	26	3rd registered blocking point	Yes, and by application
“	30		
“	31	4th registered blocking point	Yes, and by application
“	35		
“	36	5th registered blocking point	Yes, and by application
“	40		
“	41	6th registered blocking point	Yes, and by application
“	45		
“	46	7th registered blocking point	Yes, and by application
“	50		
“	51	8th registered blocking point	Yes, and by application
“	55		

Motor Identification example

To determine the following parameters, several tests are ran, the motor being disconnected from the window-lift. Carried out tests lead us to obtain:

- R: Equivalent resistor of the armature
- L: Equivalent self-inductance of the armature
- f: Viscous frictions
- J: inertia
- K: Torque constant (Kt) and speed constant (Ke), et K = Ke = Kt.

To determine R, the motor is supplied so that it doesn't start (to have the back induced electromotive force E = f (rotation speed) = 0). So we measure:

$$R = U/I = 1{,}1V/0{,}7A = 1{,}43\Omega$$

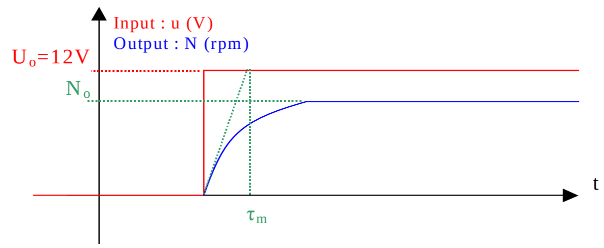
Then, the motor is supplied under nominal voltage level (Uo = 12V) to deduce permanent operation: Io = 1.2A, No = 4000 revolutions/min (rpm).

$$K = Ke = \frac{U_o - RI_o}{\omega} = \frac{12 - 1.43 * 1.2}{\frac{2\Pi * 4000}{60}} = 0.0245 \ V.s/rad$$

$$f = \frac{K * I_o}{\omega_o} = \frac{0.0254 * 1.2}{\frac{2\pi * 4000}{60}} = 65.4 * 10^{-6} Nms$$

The speed step response is analysed (to obtain mechanical parameters):

Figure 5. Motor step response analysis (nominal voltage)



We obtain:

$$Km = \frac{\omega_o}{U_o} = \frac{4000 * 2\pi}{12 * 60} = 34.9 \frac{rad/s}{V}$$

$$\tau_m = 44ms$$

and:

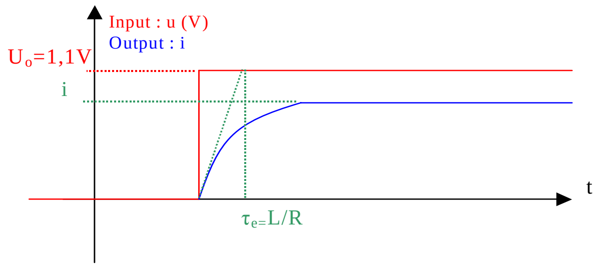
$$Km = \frac{K}{K^2 + R * f} = \frac{0.0245}{0.0245^2 + 1.43 * 65.4 * 10^{-6}} = 34.9 \frac{rad/s}{V}$$

We are able to determine:

$$J = \frac{\tau_m * K^2}{R} = \frac{0.044 * 0.0245^2}{1.43} = 19.1 * 10^{-6} Kg.m^2/s^2$$

To determine L, the motor is supplied with a low voltage level square signal so that the motor can't start. That allows to determine the electrical time constant. We measure current with a shunt resistor.

Figure 6. Electrical time constant measurement



Motor parameters can be determined as in the following script example.

```
//   Script: DC motor model:
//////////////////// Measured values ////////////////

Uo=12; // Volt
Io=1.1;   // Amp
N=4000;   // tr/min
R=1.43   // Ohms
Tau_m=44e-3;   // Mechanical constant (Seconds)
Tau_elec=0.5e-3; // Electrical constant (Seconds)
///////////////////////////////////////////////////////
wo=N*2*%pi/60;
K = (Uo-R*Io)/wo   // electromotive force constant
// Remark: We have chosen K=Ke=Kt (KE is the motor electromotive
force constant, and Kt is the motor torque constant.

f= K*Io/wo
// Using a first order model:
Ks = K/(K^2+R*f)
Km= wo/ Uo       // (rad/s)/V = Ks
J=Tau_m*K^2/R
L=Tau_elec*R
```

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

7678A–AUTO–11/06