Control of a slide flute: a mechatronic project

Brigitte D 'Andréa-Novel, Cyril Joly, Tony Noël, Joël Senpauroca

Control of a slide flute: a mechatronic project

B. d'Andréa-Novel, C. Joly, T. Noël, J. Senpauroca
MINES ParisTech-Center for Robotics - PSL Research University
60 Bvd Saint-Michel - 75006 Paris
brigitte.dandrea-novel@mines-paristech.fr
(+33)1 40 51 90 94

ABSTRACT

In this paper, we consider the problem of controlling a slide flute: a kind of recorder without finger holes but ended by a piston mechanism to modify the length of the resonator. From a physical point of view, stopped pipes have not been studied so widely as open pipes, and moving boundary conditions introduce interesting mathematical problems. To control dynamical systems, it is important to elaborate a realistic model, so that control laws can be tested efficiently before they are implemented on real size prototypes. This dynamical model has been elaborated in a previous work and the control law has been validated on a first prototype. The feedback term of this control law used on-line measurements on the system: the length of the piston through an encoder and the blowing pressure through a pressure sensor. But the flute moved too slowly. To improve the precision of the control and the velocity of the piston we have developed a new "mechatronic" prototype in our laboratory which is the object of the present paper.

Keywords

Slide Flute; Mechatronics; Control; Musical and Creative Robotics

1. INTRODUCTION

Slide flutes are mostly used for jazz and popular music, even if they sometimes appear in the classical orchestra in works like the opera by Maurice Ravel, *L'Enfant et les Sortilèges*. We are interested in this paper to control this kind of instrument made of a cylindrical stopped resonator similar to a stopped organ pipe and of a blowing mouthpiece analogous to that of a recorder. Contrary to flutes, organ pipes and recorders, the variation of the pitch is here obtained through a piston mechanism. The dynamical model we have elaborated in a previous work [1, 2, 3] took into account the nonlinear coupling effects between the jet and the pipe which was considered as a linear acoustic resonator. A modal analysis has then been performed from this model, using the linearized boundary conditions, to compute the suitable blowing pressure and the suitable pipe length to obtain a desired pitch, which consituted the "feedforward" part of our control algorithm. The feedback part was then elaborated as suitable PID-controllers to regulate the system to the desired set-point, using on line measurements on the system: the length of the piston through an encoder and the blowing pressure through a pressure sensor. The length of the resonator (equivalent to the length of the piston) was regulated through an electric DC motor, and the input blowing pressure regulated through electric servo-valves. The experimental results obtained with this first prototype were satisfying (see the video on `http://dl.dropbox.com/u/2754721/Fl%C3%BBte%20qui%20joue.avi`), but the piston velocity was not sufficiently high to produce a realistic musical interpretation as explained in [1, 2, 3].

This is the reason why we have decided to build a new version of the controlled slide flute prototype, equipped with a new linear electric motor.

In the next section we briefly describe the slide flute prototype in development in our laboratory. Then we present the different elements in more details:

- The artifical mouth
- Pressure control
- Electronics
- The electric linear motor driving the piston
- Graphical interface and operating software
- Calibration of the flute
- Experimental results

Figure 1 shows the global synoptic of the project. Each element which is described in the following sections can be found on this figure.

2. THE SLIDE FLUTE PROTOTYPE

The prototype which is called *Flutronic* is described at Figure 2. All the elements are placed in a compact plexiglass box which admits as inputs an air flow directly from a compressor, a general power supply of 220V and two USB connections for an Arduino electronic card and the motor control. The flute is made of a cylindrical body and a mouthpiece both in black ebonite. The piston of the slide flute can be translated by an electric DC motor through a slipper guide.

Concerning the excitation pressure, the system has been equipped with an air compressor supplying a proportional electric servo-valve which allows the air-regulation of the artifical mouth in silicone which has been connected to the flute to avoid air leaks. The proportional control is realized through an electronic Arduino circuit which can deliver a variable voltage between 0 and 5V thanks to a *Pulse Width Modulator* (PWM) output which has a frequency of 980Hz.

A small video showing the mounting of the virtual prototype of the flute can be previewed on the following url link: `https://www.youtube.com/watch?v=QjXP3FrRFxQ&feature=youtu.be`.

A graphical interface has been developed. It allows to define the series of notes the flute has to play and their duration.

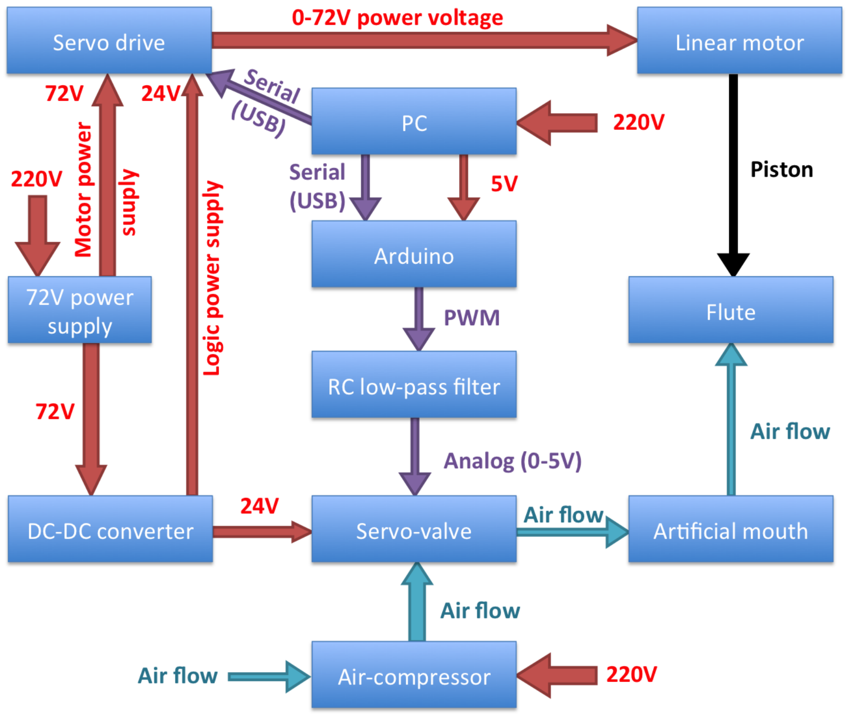

Figure 1: Synoptic diagram of the system — Red stands for power supply, purple for logic and cyan for air flow.

3. THE ARTIFICIAL MOUTH

The main challenge of this piece is the seal and the issuance of an air flow less straight as possible to promote the turbulence at the entrance of the flute resonator. The idea to produce a molding containing the end of the flute (to simulate the lips) and the air supply pipe was used. A solution with a silicone material has been chosen since it achieved the sealing goal. Moreover, to improve the directivity of the air jet, a plastic inlet chamber has been added so that the flow arrives perpendicularly, then is stirred into the chamber, passes through the lips and finally reaches the entrance of the resonator. One last piece to fix the mouth was printed in 3D. The artificial mouth is displayed in Figure 3.

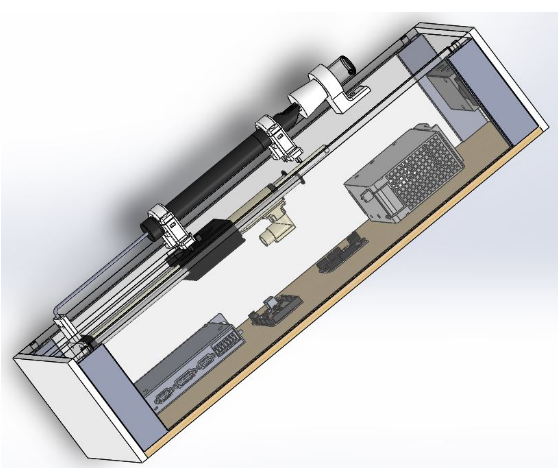

Figure 2: The new slide flute prototype

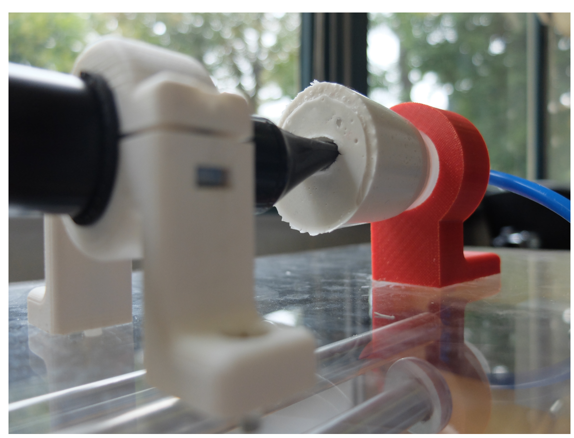

Figure 3: The artificial mouth

4. PRESSURE CONTROL

The Flutronic is supplied with air by a compressor supplying a constant pressure air stream of about $4bar$. To regulate the airflow entering the mouthpiece of the flute, we use a proportional servo valve acting as a control voltage. For that purpose, the proportional valve control device Festo type MPYE-5-420-B has been chosen to vary the direction of the air flow (in the mouth of the flute or in an exhaust pipe) proportionnally to the setpoint, sending an analog signal continuously changing the air flow. The system is powered by a $24V$ DC voltage produced by a voltage convertor Mouser 495-TEP-100-4815, and is controlled by a reference voltage varying between $0V$ and $5V$. $0V$ corresponds to a zero air flow through the mouth (and a maximum flow through the exhaust pipe) and $5V$ for a maximum air flow value by the outlet mouth (and a flow rate null by the exhaust pipe). It is then to generate a variable voltage from $0V$ to $5V$ modulating the opening of the shutter and thus controlling the loudness according to the instructions. For this, we use an Arduino circuit which can deliver a variable voltage between $0V$ and $5V$ (see next section).

5. ELECTRONICS

The Arduino card is shown in Figure 4. It is connected via two pins. The USB hub allows the module supply and delivery instructions through a serial port. The output pin delivers to the servo valve a desired voltage between 0 and $5V$ corresponding to the control set point. This can be done thanks to the *Pulse Width Modulator* (PWM) output of the card which delivers a high frequency square wave (0V and 5V) which duty cycle controls mean voltage. However, the carrier frequency provided by the Arduino PWM is only about 980Hz; this frequency is not filtered by the servo valve which results in audible variations. Thus, an analog low-pass filter based on an RC circuit was used to directly smooth the signal of the PWM. Values of resistor and capacitor are chosen to fit two constraints:

1. It has to correctly cut the 980Hz variations.

2. It should not cut desired normal variations of flow. This is directly linked to the rhythm of the music and should not exceed 10Hz.

According to these constraints, the values of $R = 680\Omega$ and $C = 10\mu$F were chosen, leading to a cut frequency $f_c = 23$Hz. Arduino includes an internal memory on which is loaded the previously written code. The instructions are delivered in "real time" as a sequence of the shutter's degrees of aperture leading to a desired sound intensity. Thus if Arduino receives instruction 0 via the serial port, it delivers a zero voltage involving the complete closure of the valve plug and no sound is provided by the flute.

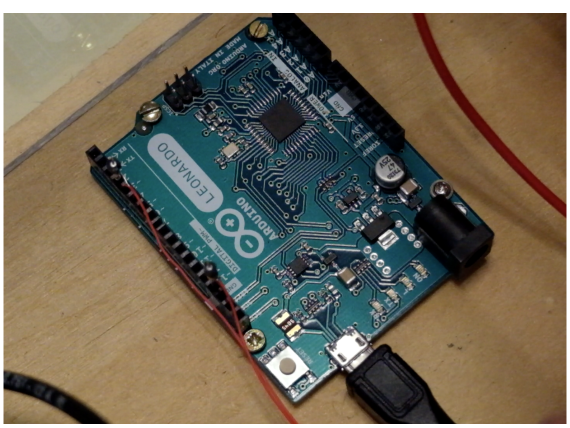

Figure 4: The Arduino card

6. THE ELECTRIC LINEAR MOTOR DRIVING THE PISTON

To move the piston of the flute, we have decided to use a **Linear Motor** instead of using a "classical" motor with a mechanical transformation of rotation in translation. We used the LinMot HS01 23x85 because it fitted very well with the mechanical constraints of the flute: it was able to move very quickly the slider of the flute. Basically, it is made of five main parts (see Figure 5):

1. The stator which contains self inductances which can generate magnetic field.

2. The slider which is made of a permanent magnet whose a part is levitating inside the stator.

3. The linear guide which is made by two metallic sliders (which are not part of the motor itself) ensuring that no radial effort may be applied on the magnetic slider (otherwise, the levitation is not guaranteed, which could add friction on the slider and damage it).

4. The DC power supply of 48V which has to be able to deliver enough current[1]

5. The Servo drive with its embedded electronics that controls the voltage to the stator. It is directly linked to the power supply and the stator and adjust the voltage to the stator depending on the command. The command is received directly through a serial port controller.

The LinMot HS01 23x85 motor is a linear motor which allows translations with low reaction times. Acceleration and deceleration will be set for our project to $8m/s^2$. To spare the slide flute, the maximum speed engine displacement must not exceeding $1m/s$. The software-hardware communication control is via a serial port controller, itself managing the servo motor position with an internal PID. The baud rate used is 57600 bits per second. Here are the main orders:

- **Off**: Disable the motor position regulation and the engine is free on its axis. Before each start, it is necessary to disable the motor a few instants.

- **Start**: Enables the motor position regulation.

- **Homing**: Homing is essential for accurate measurements. After each tensioning, the engine needs to know its position and to define its absolute zero (Figure 6).

- **Read**: This command is a query engine: "What is your current state?" The motor returns its status (moving, error) and its position. It is also used to read whether the home is complete.

- **Go to position**: This order is to move the motor in a desired set position. In addition to the position, this command has three parameters: maximum speed, maximum acceleration and maximum deceleration. The position, the velocity, the maximum acceleration/deceleration are coded on four bytes, the least significant byte first.

Each message is a sequence of a variable number of bytes. It consists of a header and a content, a $10ms$ interval is left between sending the request and reading the response. For more details, the reader can refer to the LinMot documentation [8].

[1] The current is proportional to the force applied by the motor. On this aspect, Linmot motor can be compared to rotational DC motors for which the current is proportional to the torque.

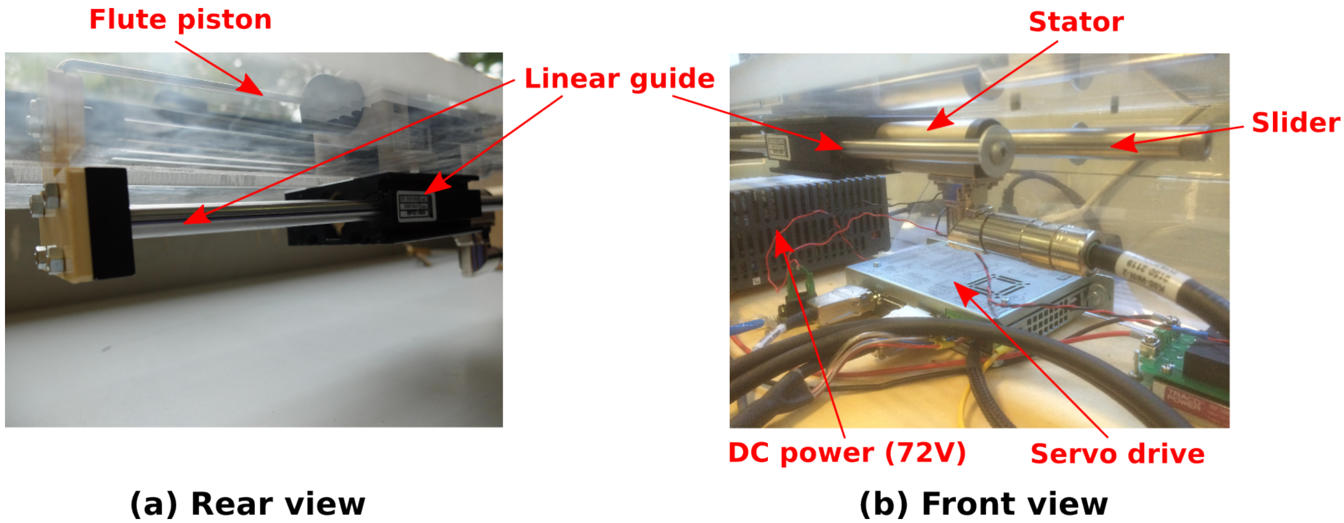

Figure 5: The LinMot

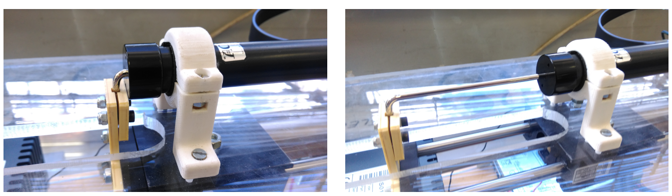

Figure 6: Left: position of the piston at the origin — Right: position of the piston at 10cm

7. GRAPHICAL INTERFACE AND OPERATING SOFTWARE

The interface provides several features (Figure 7): calibrating and tuning of the flute, loading the score's writing software, importing a given score and choosing some parameters such as the tempo and the intensity level. The tuning starts as soon as the button "tuning" is pressed. The "import a score" button opens a dialog that offers to browse the computer's files. We have chosen the musical langage "ABC" to define the musical piece we want the flute to play. In the 1980s, Chris Walshaw introduced the notation ABC to produce a textual description of the score. This allowed to produce in an easy way a MIDI file and an associated musical score as a Postscript or Pdf file from the ABC file. The information included in an ABC file contains a header and a body of music notation. An example of ABC code is as follows:

```
X: 1
T: In the moon light
L: 1/8
CCCD E2 D2 | CEDD C4 | CCCD E2 D2 | CEDD C4 |
```

In that file, X denotes the piece number, T is the title, L represents the reference length of the note. The value 1/8 specifies that the reference note is the eighth. The processing of the last line of code is used to send orders to position the linear motor to the frequency of the desired notes the flute has to play. Moreover, it also provides some information about the notes' duration and the transitions between them. This information is used by the Arduino processor to vary the air flow rate at the output of the proportional valve. The note names are replaced by the corresponding Anglo-Saxon letters: *La* is coded A etc... To move to the higher octave the notes are coded lowercase letters: a, b, c, d, e, f, g. To access the octave even higher, add an apostrophe to the right of the note lowercase letter (for example: a ', b' ...). The lower octave to the first reference is encoded with uppercase notes accompanied by a comma (eg: A, ...). The silences are encoded with the letter z and the duration of a note is represented by a digit, multiple of L, following the letter. For more details, the reader can refer to [7].

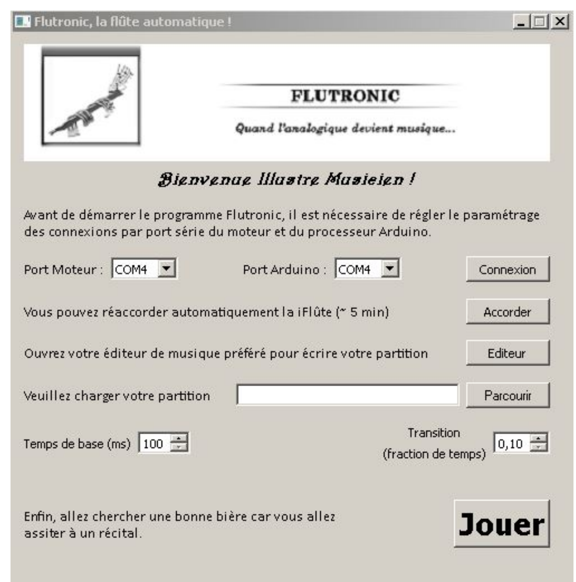

Figure 7: The musical interface — From top to bottom: The first line of buttons is used for communication setup – The second line is used for starting the calibration process – The third line is used for launching the score's writing software – The fourth line is used to load a score – The fifth line is used for setting tempo and time transition of playing – The last button is used to launch the play.

8. CALIBRATION OF THE FLUTE

To avoid the computation of a "feedforward" control, a table of correspondance between the notes played by the flute and the motor positions is elaborated. The range of the flute is from $F\sharp$ to the G of the higher octave (denoted *g*). Therefore, the table contains 14 values of motor positions corresponding to the following notes:

```
1: F# - Gb
2: G
3: G# - Ab
4: A
5: A# - Bb
6: B
7: c
8: c# - db
9: d
10: d# - eb
11: e
12: f
13: f# - gb
14: g
```

The table of notes is obtained thanks to the two following steps:

1. The motor is driven for the zero position (corresponding to the piston fully inside the flute), up to the length of the flute by steps of $3mm$. For each step, a short recording is automatically done thanks to a microphone and the signal is analyzed using a classical Fast Fourier Transform in order to get its frequency. Thus, we obtain a curve of frequency with respect to motor position which has to be stricly decreasing. So, the function of frequency w.r.t. position should be bijective.

2. Then, from the curve and the bijective property, we can associate to each note a precise position using a linear interpolation between the measured points.

The motor positions are accurate to the tenth of a millimeter.

9. EXPERIMENT RESULTS AND PERSPECTIVES

First experimental results have been obtained. In Figure 8 the first step of the calibration is shown. It may be pointed out that the system cannot produce realistic sounds for frequencies greater than $640Hz$. This is probably due to the difficulty to provide an input airflow adapted to high frequency sounds of the flute. Moreover, Table 1 shows the results of the second step of the calibration (correspondences between notes and motor positions). It can be noticed that the 3 highest notes cannot be correctly detected because of the later problem.

A first video can be seen at the url **https://www.youtube.com/playlist?list=PLC2z1KcNOkHTrYu4b1r-M8ZhEizozLVUa**. Let us also notice that when playing connected connected notes, there is no stop of the input airflow and no transition time between the two notes.

In a future work, the control of the pressure could be improved through the elaboration of an artificial mouth as it has been done for example for brass instruments [5], in the case of a trombone [4] and in the case of a trumpet [6]. We also plan to produce MIDI files in place of ABC files to be able to precise the desired intensity levels and to load any standard MIDI file as an input file.

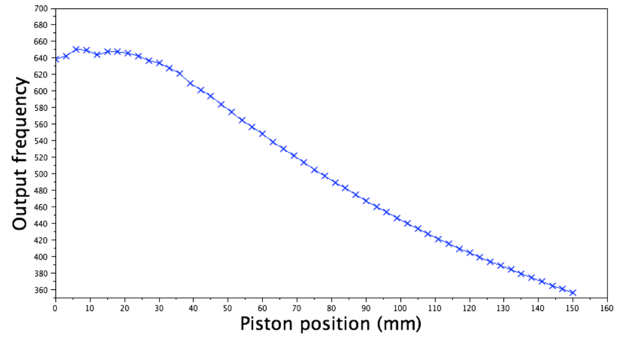

Figure 8: Experimental results

Note number	Motor position	Observation
1	143*mm*	Good value
2	129*mm*	Good value
3	116*mm*	Good value
4	104*mm*	Good value
5	93*mm*	Good value
6	82*mm*	Good value
7	71*mm*	Good value
8	60*mm*	Good value
9	49*mm*	Good value
10	38*mm*	Good value
11	18*mm*	Good value
12	48*mm*	Spurious value
13	80*mm*	Spurious value
14	114*mm*	Spurious value

Table 1: Motor position with respect to the notes

10. REFERENCES

[1] B. d'Andréa-Novel, B. Fabre, J.-M. Coron, "An acoustic model for automatic control of a slide flute", *Acta Acustica united with Acustica*, Vol. 96, pp. 713–721 (2010)

[2] B. d'Andréa-Novel, B. Fabre, J.-M. Coron, "An Acoustic Model to Control an Experimental Slide Flute", *Proc. of the CFA'12, Acoustics 2012*, Nantes, April (2012)

[3] B. d'Andréa-Novel, B. Fabre, P. Jouvelot, "Acoustique-Informatique-MusiquE - Outils scientifiques pour la musique", *Presses des Mines*, Paris (2012)

[4] N. Lopes, T. Hélie, R. Caussé, "Control of an Artificial Mouth Playing a Trombone and Analysis of Sound Descriptors on Experimental Data", *Proc. of SMAC: Stockholm Music Acoustics Conference*, pp. 521–528, (2013)

[5] J.-F. Petiot, F. Teissier, J. Gilbert and M. Campbell, "Comparative analysis of brass wind instruments with an artificial mouth: First results", *Acta Acustica united with Acustica*, Vol. 89, no. 6, pp. 974–979, (2003)

[6] C. Vergez, X. Rodet, "Experiments with an artificial mouth for trumpet", *Proc. of ISMA: International Symposium of Music Acoustics*, Leavenworth, Washington State, USA, pp. 153–158, June (1998)

[7] The ABC music standard 2.1 in **http://abcnotation.com/wiki/abc:standard:v2.1**, december (2011)

[8] LinMot - User Manual LinRS Interface in **http://www.linmot.com/fileadmin/user_upload/Downloads/software-firmware/servo-drives/linmot-talk-6/Usermanual_LinRS_e_recent.pdf**, (2014)