

THE DEVICE FOR INERTIAL SENSORS TESTING

David Skula, Jiří Fialka

Doctoral Degree Programme (3, 2), FEEC BUT

E-mail: xskula00@stud.feec.vutbr.cz

Supervised by: Luděk Žalud, Petr Beneš

E-mail: {zalud, benes}@feec.vutbr.cz

Abstract: The article deals with hardware and software design of the device for inertial sensors testing. The device is designed as a two-axis rotation system. One axis is in horizontal plane and one in vertical plane. Visualization and a control interface are built in LabVIEW and control drivers in C# .NET 3.5.

Keywords: inertial sensors, testing, LabVIEW, C#

1. INTRODUCTION

Inertial sensors are accelerometers (for acceleration measurement) and gyroscopes (for angular rate measurement). Last few years are popular sensors based on MEMS technology, which allows creating mechanical parts of sensors (referential mass of gyroscope and accelerometer) on silicon chip with dimensions of the order of millimeters. The accuracy of these sensors is not very good yet. The MEMS sensors low accuracy is a reason why we need a device for testing these sensors. If we are able to quantify a sensor error then we can compensate it.

2. MECHANICAL CONSTRUCTION

The mechanical construction is in design phase at this moment.

A sketch of mechanical design is shown in Figure 1. The device is designed as a two-axis rotation system. One axis is in horizontal plane and one in vertical plane. This concept is compromise between an ideal three-axis rotation system and mechanical complexity and cost.

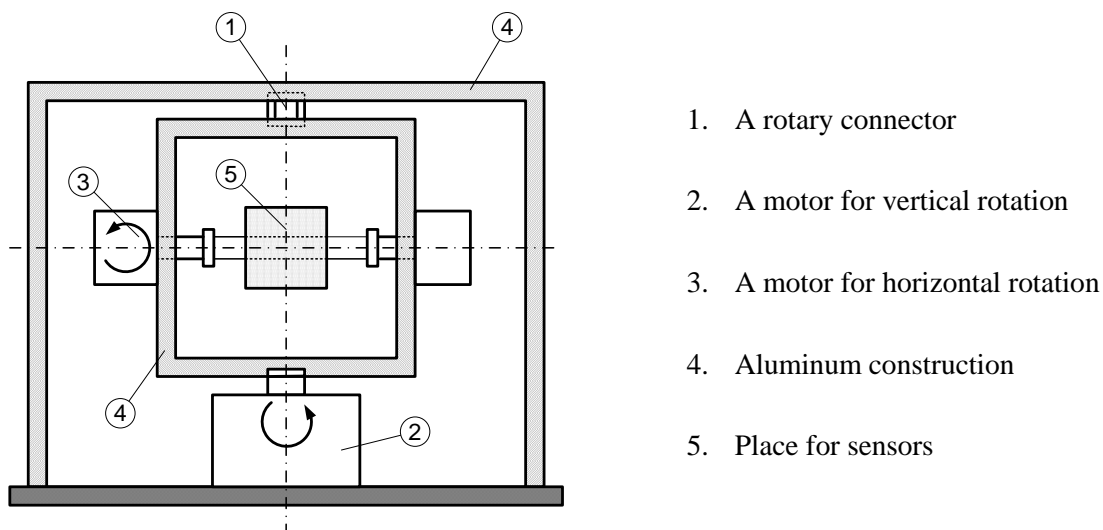


Figure 1: A sketch of mechanical design

Whole construction is built from aluminum profiles (4). Power to the second motor (3) is conducted through a rotary connector (1). This allows continuous rotating in vertical plane. Sensors are battery powered and placed on a mat (5). Measured data are transmitted wirelessly to a PC. This allows continuous rotating in horizontal plane.

Motors are from TG Drivers, s.r.o. and are controlled by digital servocontrollers from the same company. Types of motors are not selected yet. The vertical motor will be equipped with a precision encoder. This allows continuous rotating with precision rotation speed monitoring. The horizontal motor will be used to setting the horizontal inclination angle. This arrangement allows testing and comparing various sensors.

3. SOFTWARE

The control software is built in LabVIEW and C# .NET 3.5. Situation is shown in Figure 2.

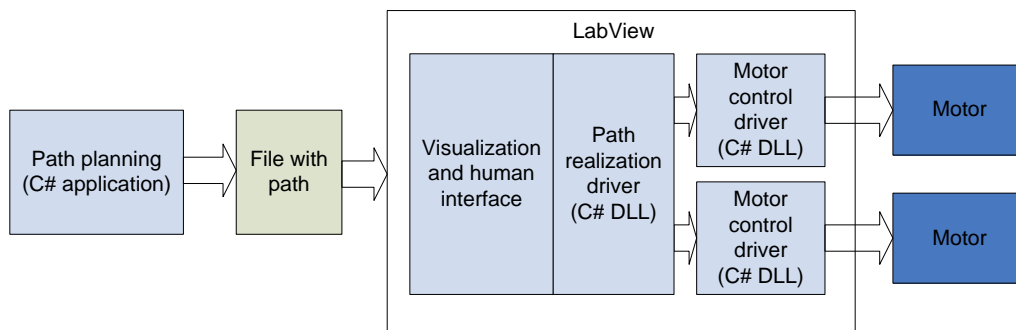


Figure 2: The software block diagram

The software is divided into four parts:

- A path planning application.
- Drivers for motor control.
- A driver for path realization.
- A visualization and human interface.

3.1. A PATH PLANNING APPLICATION

Path planning application is built in C# .NET 3.5 and is used for motor movement planning (Figure 3). A user is able to build a set of movement for each motor separately. Three variants of motor control are available:

- An absolute position control.
- A relative position control.
- An angular rate control.

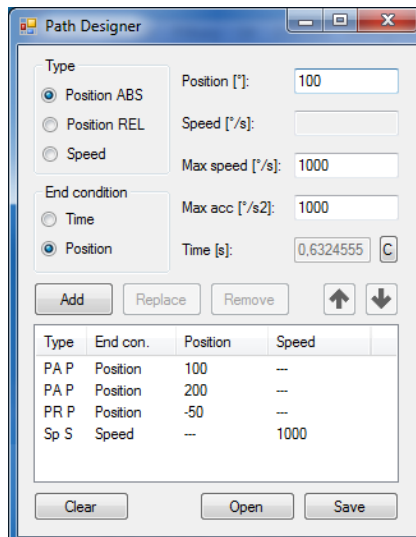


Figure 3: The path designer application

The absolute position control is the most using one. Basics parameters are a destination angle (“*Position*”), a maximal angular rate (“*Max.speed*”) and a maximal angular acceleration (“*Max.acc*”). A condition for the moving from the current point o path to the next point is selectable in section “*End condition*”. The “*Position*” means that the program ends the actual point immediately after the entered position is reached. The “*Time*” means that the program ends actual point after the specified time. The motor waiting in destination position until the specified time is reached.

The relative position control works as well as the absolute position control with a one difference. The “*Position*” is not a destination angle but a relative rotation angle (angle from an actual position to a destination position).

The angular rate control is different than the previous two. Basic parameters are a destination speed (“*Speed*”), a maximal angular rate (“*Max.speed*”) and a maximal angular acceleration (“*Max.acc*”). The end conditions are a destination speed or a specified time.

These three modes can be combined. A path plan is saved to a file at the end.

3.2. DRIVERS FOR MOTOR CONTROL

Communication with servocontrollers is established trough RS232. Communication protocol is difficulty implementable to LabVIEW. This is a reason why motors drivers are built in C# .NET 3.5 like DLL libraries. These are implemented in visualization software in the LabVIEW. Drivers include all the basic functions for motor control (set destination position, maximal speed, maximal acceleration or read position, speed, status, etc.).

The drivers are built for an easy implementation to the LabVIEW. Using of the drivers is divided into three parts:

1. Required functions registration.
2. Start of the driver.
3. Control of the motor (using of the registered functions).

The drivers are initialized firstly. Then functions that will be used are registered (set destination position, maximal speed, maximal acceleration or read position, speed, status, etc.). The registered functions can by later used for communication with the motors only. Each function can be registered as cyclic or single. Cyclic functions are called in each driver cycle. Single functions are called

if specified condition is satisfied. This condition is parameter change for writing functions and manual calling for reading functions.

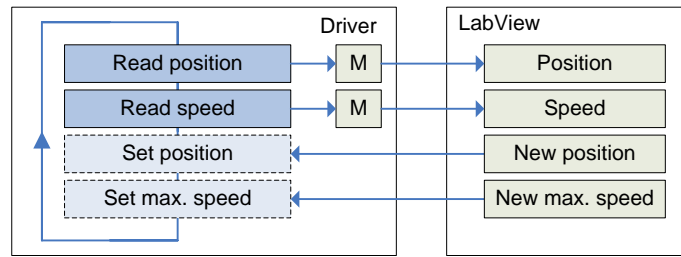


Figure 4: A motor control example

The driver is then started. All registered functions marked as cyclic are called in each driver cycle. All registered functions marked as single are called at specified condition. An example is shown in Figure 4.

Example: Registered cyclic functions are “*Read position*” and “*Read speed*”.

Registered single functions are “*Set position*” and “*Set maximal speed*”.

The driver reads in each driver cycle an actual position and an actual speed and saves these to the memory (M). If user changes destination position, the “*Set position*” function is called once. If user changes the maximal speed, the “*Set maximal speed*” function is called once.

This way is suitable for using in LabVIEW. Visualization is not depending on communication timing.

3.3. A DRIVER PATH REALIZATION

Driver for path realization is built in C# .NET 3.5 like DLL library and is implemented in visualization software in the LabVIEW. This driver loads path from file (path planning application) and realizes path by using motor control driver.

The path realization driver contains three input parameters:

- A file with path.
- Actual position.
- Actual speed.

The actual position and the actual speed are feedback for condition for next point of path and can be directly link to the corresponding cyclic read functions in the motor control driver.

The path realization driver contains five output parameters:

- A mode of motor (absolute position control, relative position control, speed control).
- Destination speed (speed control).
- Destination position (position control).
- Maximal angular speed.
- Maximal angular acceleration.

The output parameters can be directly link to the corresponding write functions in the motor control driver.

3.4. A VISUALIZATION AND HUMAN INTERFACE

The visualization and human interface for device settings are built in LabVIEW. This application contains the motor control driver and the path realization driver. A visual style of this application is not complete yet.

4. CONCLUSIONS

The mechanical part of the device is in design phase at this moment. The software is done. Whole device will be used for testing and comparing of inertial sensors.

A user can build a test path in the Path designer and then used it for testing multiple sensors. Sensors are tested under the same condition in such case and the results are simply comparable.

ACKNOWLEDGEMENT

This work has been supported in part by Ministry of Education, Youth and Sports of the Czech Republic (Research Intent MSM0021630529 Intelligent systems in automation), Grant Agency of the Czech Republic (102/09/H081 SYNERGY - Mobile Sensoric Systems and Network) and by Brno University of Technology.

This work was supported by grant „Modern Methods and Approaches in Automation“ from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-10-12).

REFERENCES

- [1] Albahari J., Albahari B.: *C# 4.0 in a Nutshell*, O'Reilly Media 2010, ISBN 978-1449380458
- [2] Valach, J., Havlíček, J., Valach, M.: *Začínáme s LabVIEW*, BEN 2008, ISBN 978-80-7300-245-9
- [3] Titterton, F. H., Weston, J. L. *Strapdown Inertial Navigation Technology*. Institution of Engineering and Technology, 2004. 581pages, ISBN 1-56347-693-2.
- [4] Lawrence, A. *Modern Inertial Technology*. Springer-Verlag, 1998. 278 pages. ISBN 0-387-98507-7.
- [5] Woodman, O. J. *An Introduction to Inertial Navigation*. University of Cambridge, 2007, 37 pages, ISSN 1476-2986.