Validation of Yaw Damper Controller via Hardware in the Loop Simulation

Libor Waszniowski[1] and Zdeněk Hanzálek[2]
Department of Control Engineering, Czech Technical University, Faculty of Electrical Engineering, Karlovo nám. 13, 121 35 Prague 2, Czech Republic

and

Jiří Doubrava[3]
AERO Vodochody a.s., U Letiště 374, 250 70 Odolena Voda, Czech Republic

The paper presents a validation of the Yaw Damper Controller with hydraulic actuators via hardware in the loop simulation on a stand with a revolving platform and a loading force actuator. The Yaw Damper Controller is briefly described and requirements on validation tests are mentioned. An architecture and software and hardware components of the developed simulator and the stand are described. The simulator instrumentation is based on distributed peripherals communicating via CANopen industrial field bus. The real time simulation of the plane model is performed via an embedded computer with Power PC processor and Linux operating system. Visualization and a data evaluation are provided via a graphical user interface in the Matlab environment. Validation test proving the Yaw Damper Controller software error is presented.

Nomenclature

YDC	=	Yaw Damper Controller
FOG	=	fiber optic gyro
HIL	=	hardware in the loop
GUI	=	graphical user interface
SW	=	software
HW	=	hardware
SA	=	serial actuator
PA	=	parallel actuator
FS	=	force sensor
Ctrl	=	controller
LVDT	=	linear variable differential transformer
ECU	=	electronic control unite
Cfg	=	control panel
Q	=	dynamic pressure
HMI	=	Human Machine Interface
SimFS	=	simulator of force sensor FS
SimQ	=	simulator of dynamic pressure Q
ForceCtrl	=	simulator of aerodynamic loading force
LA	=	load actuator
SimTarget	=	computer providing real time simulation
SimGUI	=	graphical user interface of the simulator
CAN	=	controller area network
MCU	=	microcontroller
X	=	spring length

[1] Research Assistant, Department of Control Engineering, Czech Technical University, E-mail: xwasznio@fel.cvut.cz
[2] Associate Professor, Department of Control Engineering, Czech Technical University, E-mail: hanzalek@fel.cvut.cz
[3] Development Engineer, Department of Aircraft System Development, AERO Vodochody a.s., E-mail: jiri.doubrava@aero.cz

I. Introduction

The Yaw Damper Controller (YDC) is a device directly coupled with the plane rudder. It senses yaw rate via a fiber optic gyro (FOG) and compensates oscillations via hydraulic actuators deviating the rudder. Event though it is designed to be fail-safe (it is disconnected from the ruder when a failure is detect) and the direct mechanical coupling of the rudder with the pedals guarantee controllability of the plane when the YDC does not work, a malfunction of the YDC could have a serious effect on the fly safety. For example, too big deflection of the rudder in high speed of the plane can cause overloading and damage of the rudder or poorly tuned close-loop dynamic of the system can induce rudder oscillations which is particularly danger at rudder resonance frequency. Therefore, the validation of the whole subsystem controlling the ruder must be carefully done, event though it is assembled from certified components.

This paper presents a top level validation of the YDC via hardware in the loop (HIL) simulation on a stand consisting of a complete hydraulic and electronic subsystem of the YDC, the most relevant mechanical parts of the rudder and a numerical simulator of the plane dynamic.

The basic principle of the hardware in the loop (HIL) simulation is based on connection of real hardware devices (YDC with electronic, hydraulic and mechanical subsystems and rudder in our case) with a computer based simulator providing a real-time numerical simulation of the rest of the system (plane in our case) in the closed loop. The idea of connection of the real part of the system with the modeled rest of the system is based on the following arguments. Due to the presence of the real devices in the closed loop, the simulation is as near to the reality as possible in this validation phase. Due to the utilization of the model in the closed loop, the simulation is cheaper, easier and less time consuming than experiments on the real device, risks of the damage of the real device are eliminated and simulation of the system behavior in extreme modes of operation which are hard, danger or even impossible to induce on the real device is possible. The absence of the real device in the simulation particularly simplifies the development of a plane control system, since it allows systematically and effectively validates the system, even if the plane or a certified component is currently not available. A measurement and data logging implementation is much easier and flexible on a stand than on a plane. All input signals are deterministic in the computer based HIL simulation. The reproducibility of a passed experiment is therefore easy contrary to fly tests.

The objective of our work is to develop a modular HIL platform for testing behavior of an airplane control system. The utilization of this platform in the process of the YDC validation is presented in this paper. The plane yawing is simulated by rotating the gyro on revolving platform, the angle of the rudder is measured via sensor and the aerodynamic force loading the rudder is imposed via a hydraulic cylinder. The system contains an embedded computer executing the numerical model of the plane in real time and peripherals interconnected via CANopen field bus which make this architecture very flexible and extensible. The graphical user interface (GUI) is implemented in Matlab which allows a test evaluation in the same environment where the model and control law was developed. Even a very simple experiment passed on this HIL simulator has proved an error in YDC SW - the sign of the yaw rate measured by the FOG was wrongly interpreted by the YDC SW under some conditions.

The rest of the paper is organized as follow. The YDC rough architecture and basic functionality is described in Section II. This chapter also defines requirements on validation tests. Section III deals with the developed HIL simulator. It describes she simulator architecture, the stand with mechanical and hydraulic equipment and its electronic instrumentation. Also the main features of the simulator computer and the GUI are mentioned. An example of a validation test passed on the HIL simulator is described in Section IV. This test shows an error in the YDC software. Concluding remarks are mentioned in Section V.

II. Yaw Damper Controller

A. YDC Architecture

The Yaw Damper Controller is a device sensing the yaw rate of the plane via a fiber optic gyro and compensating its oscillations via hydraulic actuators deviating the rudder. The YDC actuation is additive to pilot commands. The direct mechanical linkage of pedals with the rudder is maintained and the driving force of the YDC actuators is mechanically mixed with the pilot driving force. The YDC fundamental blocks with the basic kinematics scheme are depicted in Figure 1.

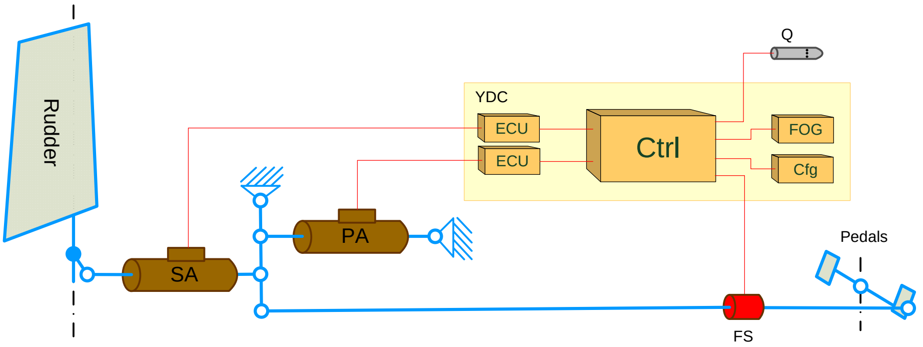


Figure 1 Kinematics schema and block diagram of the YDC

The YDC uses two hydraulic actuators (parallel and serial) to mix the YDC driving force with the pilot driving force. The serial actuator (SA) is connected in serial with pedals movement. Therefore its deflection is additive to a deflection of pedals. SA is used to add the YDC actuation needed to compensate the yaw oscillations to the rudder deflection. To prevent propagation of the SA driving force to pedals, a parallel actuator (PA), connected in parallel with the pedals movement, is used. The YDC measures the force in pedals via a force sensor (FS) and drives the PA to reach the force corresponding to the aerodynamic load of the rudder. The rudder aerodynamic load corresponds to the rudder deflection and dynamic pressure Q measured via Pitot tube. Yaw rate is measured by a fiber optic gyro (FOG). The controller (Ctrl) computes the SA deflection as the yaw rate derivation with the first order filter. The maximal SA deflection is limited according to speed. Actuators deflection is measured by linear variable differential transformer (LVDT) sensors and their closed-loop control is provide by an analog electronic control unite (ECU) – one for each actuator.

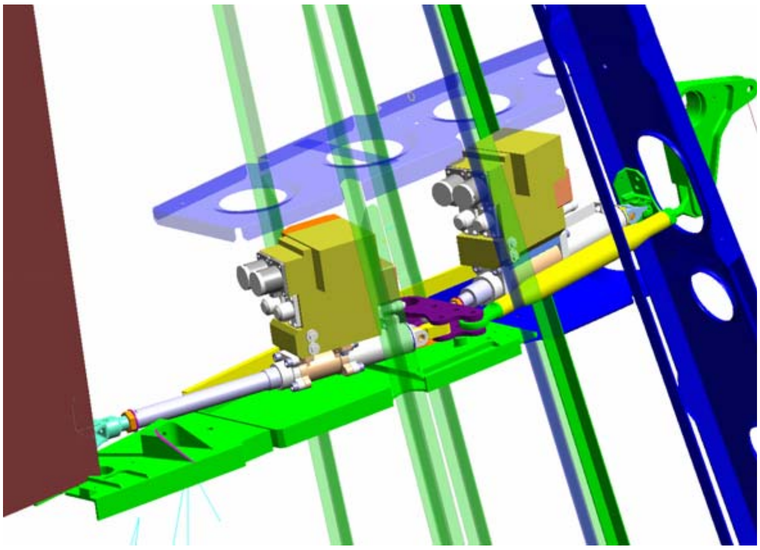


Figure 2 YDC actuators installed in the vertical stabilizer

The pilot can disengage the YDC via a control panel (Cfg). Moreover, YDC disengages itself when some fault occur (e.g. the wanted actuator deflection is not reached within a defined time), when the force in pedals exceed defined value (pilot probably tries to overpower the YDC) or when rudder oscillations near its resonance frequency are detect. When the YDC is disengaged, the SA is fixed by closing its hydraulic valves and the PA is relaxed by opening a hydraulic bypass valve. The manual control of the rudder is therefore maintained. The pilot can engage the YDC via the control panel. Than a self-test is passed and some other enabling conditions (e.g. plane is flying) must hold.

The SA and PA are installed directly in the plane vertical stabilizer (see Figure 2).

B. YDC Test requirements

To validate the YDC functionality described in Section A, several HIL test has been specified [2]. The goal of these tests is:

1. to identify the real dynamic and static characteristics of the control system
2. to evaluate the control quality via measurement of the rate of dumping of the yaw rate
3. to analyze the effect of nonlinearities in the system
4. to analyze the effect of faults in the system

These tests are carried out:

a) in several points of the flight envelope
b) for different configurations of the plain (especially for different rotational inertias)
c) off-load and for various aerodynamic load of the rudder

Identification the real dynamic and static characteristics of the control system

The dynamic model of the plane and many mechanical and hydraulic components of the control system are nonlinear. Moreover, dynamics of some components (hydraulic actuators in our case) is unknown at the design time, since these components are still under development. The control law design is therefore based on a simplified linearized model omitting or estimating unknown dynamics and nonlinearities. Even though the robustness analysis is used to verify the control law at the design time, an experimental validation of the system behavior based on real system dynamics must be done at the evaluation time.

Identification of the real control system dynamics can be done in the open loop. There are two loops in the control system. The outer loop consist of the gyro FOG, corresponding control law implemented in the computer Ctrl, the electronic control unit ECU and the serial actuator SA. The inner loop consists of the force sensor FS, corresponding control law implemented in the computer Ctrl, the electronic control unit ECU and the parallel actuator PA. Various amplitudes and periods of harmonic input signals are used to find the nonlinear behavior.

The most important issues was the dynamic and static characteristics of the new actuators, nonlinearities of the force sensor and the actuators and the control law containing a Kalman filter eliminating the drift of the FOG. Other important experiment verifies the static characteristics limiting the rudder deflection in relation to the dynamic pressure Q.

Evaluation of the control quality via measurement of the rate of dumping of the yaw rate

The rate of dumping of the yaw rate is measured in the system with the real dynamics (containing real actuators and mechanical component with reduced inertia) and real nonlinearities (friction and hysteresis of mechanical components, force sensor and actuators). Rate dumping is measured in the defined point of the flight envelope (attack ground target) where the highest control quality is required. The effect of the aerodynamic load is evaluated via simulation of this load in the HIL simulator.

The rate of dumping is measured in the closed loop consisting of FOG, Ctrl, ECU, SA, Rudder and the numerical model of the plane dynamic computing the yaw rate of the plane from the Rudder deflection. This system is excited by a FS signal which causes the Rudder deflection via PA. Various shapes of the exciting signal can be used. Typically it is a symmetric or asymmetric pulse of the sine or rectangular shape, sequence of several pulses or periodic signal corresponding to disturbances caused by gunfire. Various amplitudes and periods of the input signals are used to take the nonlinear behavior into consideration.

Analyzes of the effect of nonlinearities in the system

Backlashes, friction and other nonlinearities in the closed loops can cause undesirable oscillations of the system. These oscillations can be excited by appropriate signals during the HIL simulation. The intention is to identify the source of these oscillations. It can be the inner loop (PA, mechanical linkage, FS, Ctrl, ECU, PA) or the outer loop (SA, Rudder, plane, FOG, Ctrl, ECU, SA). The effect of the system parameters on these oscillations should be evaluated via experiments.

Analyzes of the effect of faults in the system

It is necessary to validate the operation of the system under various fault conditions. The most dangerous faults are that causing the rapid deflection of the Rudder. Various faults should be injected to the electrical system to verify that no fault can cause such behavior.

Another dangerous phenomenon is resonance of the rudder. The algorithm executed in Ctrl must identify oscillation of the system on frequency near the critical frequency of the rudder and disengage the YDC. This should be tested via induction of the critical frequency to the inner loop and outer loop. The inner loop is excited via force sensor FS. The outer loop is excited via yaw rate measured by FOG.

It should by also verified that the YDC disengage when the force on FS exceeds a limit, values of dynamic pressure Q from two independent sources differs or actuators do not reach reference position within specified time.

III. Hardware in the Loop Simulator

In this chapter we describe the HIL simulator developed for YDC tests. Architecture of a general HIL simulator is depicted in Figure 3. Sensors and Actuators provide a physical interface with the Control Unit Under Test. The model of the controlled plant is simulated by the Control and Simulation Calculation block. This block also controls

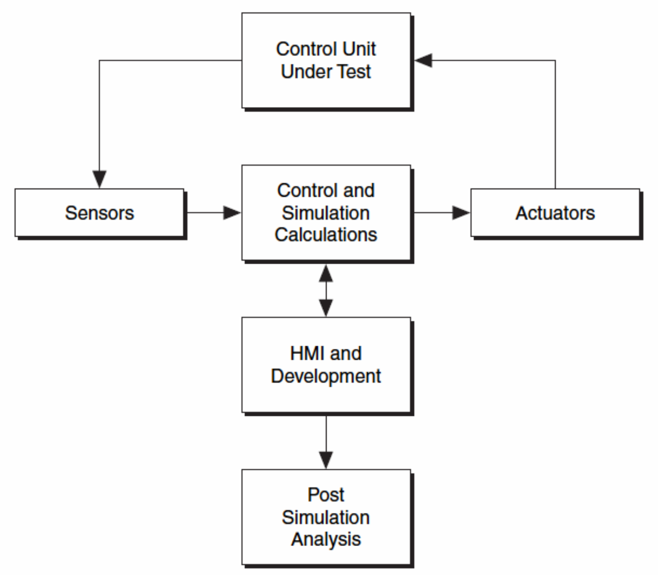


Figure 3 Architecture of a Typical HIL System 1

the data acquisition and actuation and controls the overall simulation process. The simulation is executed in real-time. The most commonly used mathematical model of the controlled plant is discrete and supposes equidistant synchronous sampling. Sensors, Actuators and Control and Simulation Calculation block must be therefore precisely timed and synchronized.

As important as closed loop simulation itself are supporting tools providing Human Machine Interface (HMI) of the simulator, development and parameterization of the model and Post Simulation Analysis. While blocks composing the closed loop (Sensors, Actuators and Control and Simulation Calculation) affect the accuracy of the simulation, the supporting tools (blocks HMI and Development and Post Simulation Analysis) affect the efficiency of the development and validation process.

There are several factors to consider when designing a HIL system [1]:

- The system should accept a variety of control unit configurations.
- A small change in the control unit must not warrant a design of a completely new system.
- The system should perform both open and closed-loop testing.
- The system should be scalable and open.
- The system should be of reasonable cost in terms of components and development time.

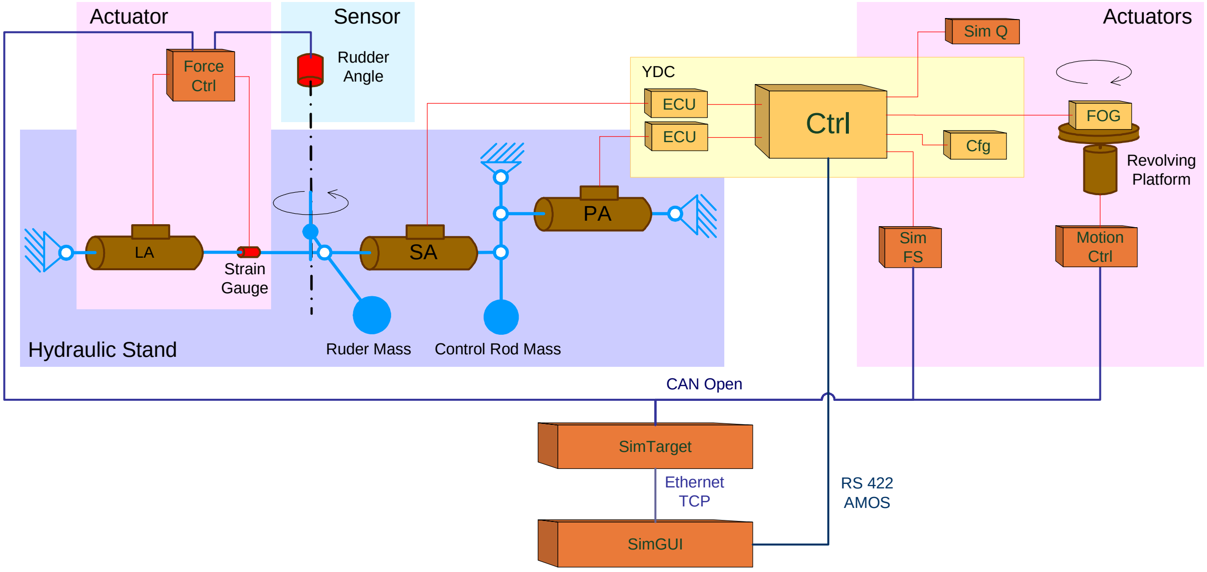


Figure 4 Architecture of the YDC HIL simulator

C. Architecture

The architecture of the developed HIL simulator for YDC tests is depicted in Figure 4. The Control Unite under Test is the YDC computer with ECUs and Paralel and Serial Actuators (PA and SA) installed on Hydraulic Stand. The only sensor (from the HIL concept point of view) is Rudder Angle sensor.

Actuation of the tested YDC is provided via Revolving Platform simulating yaw rate, SimFS simulating electric signals of the force sensor FS, SimQ allowing manually setting dynamic pressure Q and load actuator LA, installed on the Hydraulic Stand and controlled via ForceCtrl, simulating aerodynamic load of the rudder.

The real time computation of the plane dynamic model and data acquisition is provided via embedded computer SimTarget. The real time communication between sensors, actuators and SimTarget and their synchronization is provided via CAN bus and CANopen protocol (http://www.can-cia.de/).

A Human Machine Interface, model and test cases design and post simulation analysis are provided via a Personal Computer running Matlab and SimGUI tool. SimGUI gather data from SimTarget via Ethernet with simple application protocol implemented under TCP. Additional data are acquired from YDC via RS422 port with AMOS protocol intended for monitoring purposes.

Notice that the described architecture of the HIL simulator is very scalable and extensible. Many off the shelf industrial devices can be connected via CANopen – the open industrial standard (http://www.can-cia.de/).

If an appropriate off the shelf device needed for realization of an actuator or sensor does not exist, it can be developed in house since the used communication technology (CAN and CANopen) is relatively simple, opened, implemented in many microcontrollers and familiar to many HW and SW developers.

If the communication capacity of one CAN bus is not sufficient, the second port of the SimTarget computer can be used. If the computation power of the SimTarget is not sufficient, another computer can be connected to the same bus (configured as a salve).

D. Hydraulic Stand

The Hydraulic stand, depicted in Figure 5, is the central part of the HIL simulator. It consists of the real hydraulic actuators (PA and SA) and related hydraulic equipment, mechanical transmission with weights representing reduced mass of the control rod and the rudder and the hydraulic actuator LA connected with serial servo SA, which is

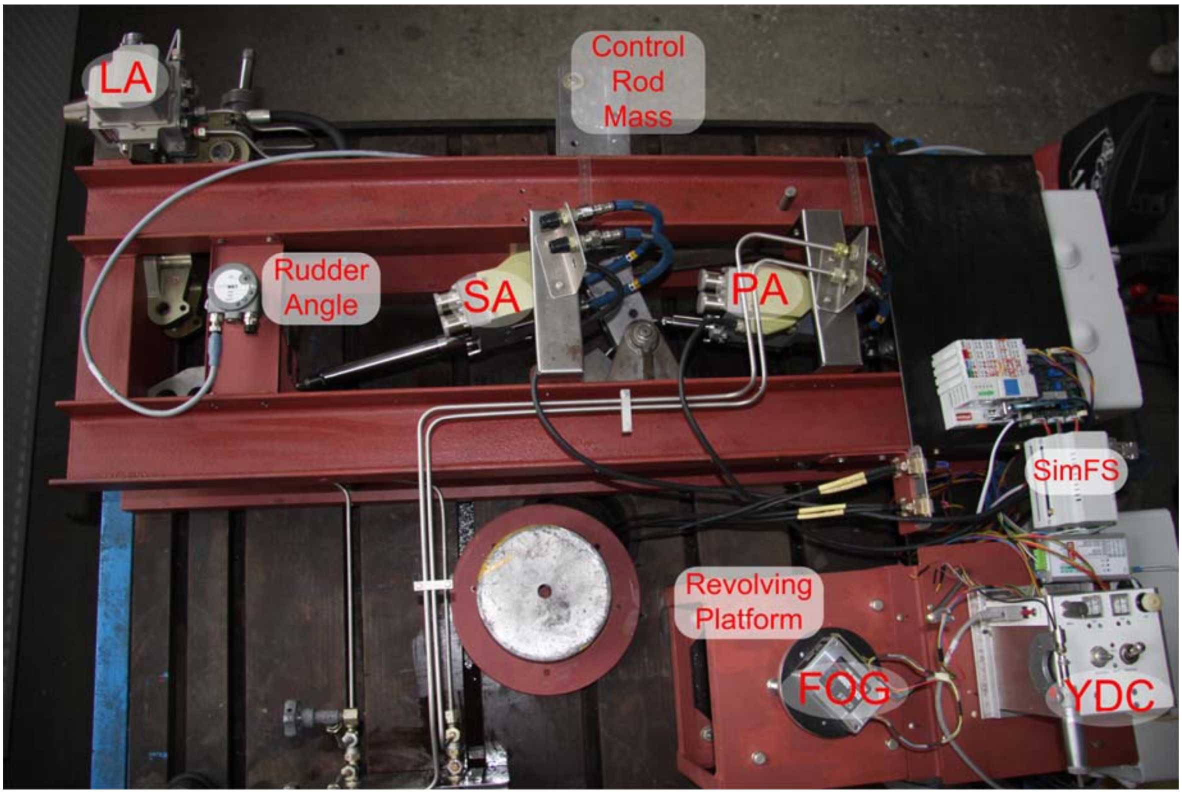


Figure 5 Hydraulic stand with instrumentation

controlled to simulate the aerodynamic load of the rudder. The hydraulic stand therefore contains all real components providing any undesirable features as a friction, backlash and other nonlinearities or unknown dynamics which has been omitted or only estimated at the YDC control law design. It is therefore the crucial part of the HIL simulator need for the system validation.

The hydraulic stand can be configured for different experiments such as measurement of static and frequency characteristics of actuators, analysis of resistance to auto-oscillations, measurement of actuators stiffness, analysis of sensitivity to reduced mass change and test of actuators in isolation and in assembly.

E. Actuators and Sensors

Rudder Angle Sensor

The rudder angle is measured via absolute single-turn optical shaft encoder with 13 bit resolution and high measurement linearity. It is an industrial device compliant with CANopen standard. It is designed for really synchronous position acquisition of several axes. Therefore it is suitable for synchronous measurement in HIL simulator.

SimQ

The pressure sensors measuring the dynamic pressure Q in the YDC Ctrl has been substituted by electric potentiometer to allow adjust Q manually. This simple solution is sufficient because it is not necessary to simulate dynamic changes of the dynamic pressure during the planed experiments. The value of Q measured by YDC is accessible via diagnostic line AMOS.

SimFS

The force sensor FS measuring the force in pedals is constructed as a Linear Variable Differential Transformer (LVDT) sensor measuring the compression of a spring caused by the measured force. The LVDT sensor is energized by harmonic voltage on primary coil. The output voltage on secondary coils connected in opposite phases is proportional to deflection of the LVDT core from its middle position (ΔX) that is the change of the spring length X.

This functionality is simulated via device SimFS (see Figure 6). It is an analog amplifier with variable gain. The gain is controlled by a microcontroller (MCU) connected to the SimTarget via CAN bus with CANopen protocol.

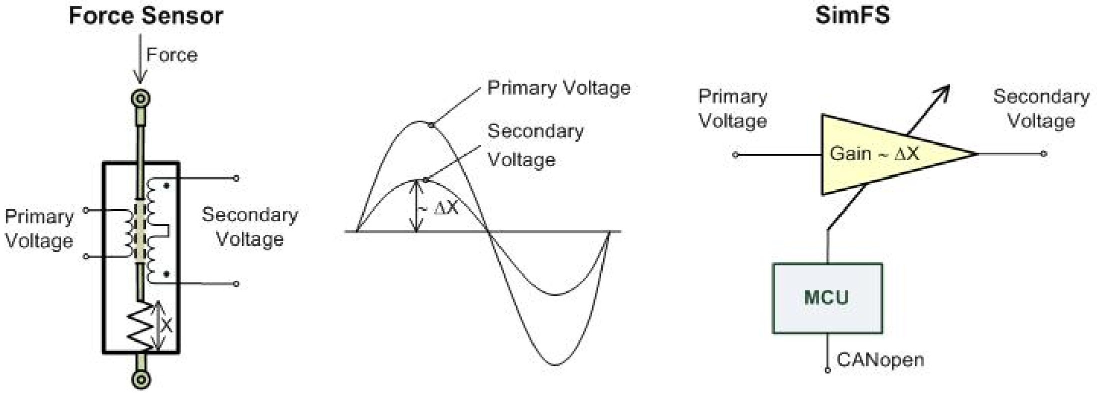


Figure 6 Schema of the Force Sensor and its simulation via SimFS

Revolving Platform and Motion Control

The yaw rate of the plane is measured by the fiber optic gyro (FOG). Kalman filter is used in the YDC control law to eliminate the drift of the FOG value. Since a mathematical model of the FOG drift is unknown, the real FOG must be used to validate the tuning of the filter. The FOG is, therefore, rotated by the revolving platform simulating the yaw rate of the plane.

There are very strict requirements on the lowest angular speed which is 0,05 deg/s. A gearbox with high transmission rate, a precisely controlled electrical servomotor and an encoder with high resolution and linearity must be used to satisfy this requirement. The planetary gearbox with transmission rate 100 has been used. The second crucial parameter of the gearbox in this application is the backlash which must be as low as possible. Our experiences from experiments show, however, that it is not necessary to use very expensive zero-backlash gearboxes. A gearbox with guarantied backlash lower than 3 degree has been sufficient in our case.

An AC synchronous servomotor has been used to drive the platform. It is a 3 phase, 6 pole permanent magnet synchronous motor with sinusoidal shape of the back electromotive force (EMF). Motor feedback is provided by an embedded encoder with 15bit resolution. The motion controller (MotionCtrl) is a digital controller tuned for the dynamics of the platform. It is connected via CANopen.

The other requirements on the revolving platform are maximal angular speed 40 deg/s and maximal acceleration 120 deg/s^2. These requirements, however, have not been crucial for components selection and the platform construction.

Load Actuator and Force Control

The aerodynamic force loading the rudder is simulated via Load Actuator (LA) consisting of hydraulic cylinder and servovalve. The servovalve is controlled by a digital controller ForceCtrl. The force is measured by four foil strain gauges forming the full Wheatstone bridge attached to the connecting rod of the cylinder. Even though an AC power supply of the bridge can be used to eliminate disturbances, the DC power supply of the bridge has been used, since the measured force can change dynamically (the frequency band of the controller has been estimated to 1kHz at the design time) and the required frequency of the AC power supply would be therefore to high.

The value of the current aerodynamic force, which is used as a reference for the loading force, is computed by ForceCtrl according to the following equation:

$$F_{AeroLoad} = k_\alpha \alpha + k_\beta \beta \tag{1}$$

where:

α	is angle of the rudder
β	is drift angle
k_α and k_β	are constants

Alternatively, the reference for loading force can be obtained from the SimTarget as a constant or time varying signal F_{ref}.

ForceCtrl is connected to two CANopen networks. It is a slave in the main network used for simulation control. This network is synchronized with the frequency of simulation (more than 12ms). ForceCtrl receives β and F_{ref} from SimTarget and transmit α and $F_{Measured}$ to SimTarget via this network. ForceCtrl is also a master of the second network used to acquire α from the Ruder Angle sensor. This network is synchronous with the ForceCtrl controller period which is 1ms.

F. SimTarget

SimTarget is an embedded computer providing data acquisition and real time computation of simulation steps. Its HW is based on the MPC5200B embedded processor with 760 MIPS Power PC core. It is equipped with 128MB SDRAM, 32MB Flash, 10/100 Ethernet, RS232, two CAN ports and many other peripherals. Its SW is based on open source components. As an operating system is used Linux 2.6, as a driver of CAN is used Socket-CAN and as an implementation of the CANopen protocol is used CANFestival (http://www.canfestival.org/).

The SimTarget architecture is depicted in Figure 7. The CANopen interface provides data exchange with sensors and actuators. LTI simulator accesses this data through an object dictionary and provides computation of simulation steps of linear time invariant (LTI) model of the plane. Inputs and outputs values are multiplied by scale constants and a reference vector form GUI interface can be added. A reference vector can be also transmitted directly to a selected actuator.

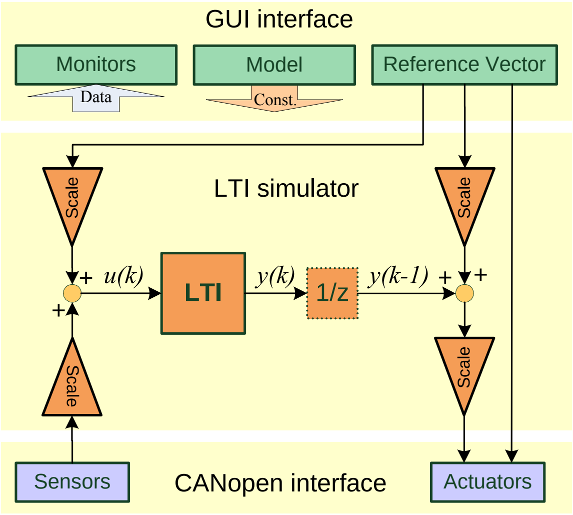


Figure 7 SimTarget architecture

The algorithm of the LTI block is defined by standard discrete state space equations:

$$\begin{aligned} y(k) &= Mx(k) + Nu(k) \\ x(k+1) &= Cx(k) + Du(k) \end{aligned} \qquad (2)$$

where:

u(k) is the input vector at simulation step *k*

y(k) is the output vector at simulation step *k*

x(k) is the state vector at simulation step *k*

M, N, C, D are matrixes of the discrete state space model

The block 1/z expresses the fact that there is a one step delay in the simulator which is caused by the simulation timing (see Figure 8). Because inputs sampling and output actuation are required to be synchronous and some time is necessary to compute and transmit data, the output *y(k)* computed from input *u(k)* is used in the next simulation step *k+1*. The simulation steps times are defined by the synchronization message SYNC transmitted by SimTarget via CAN. Sensors measure their signals *u* at this time instance and transmit these current values immediately. Actuators receive signals *y* at any time during the simulation period, but they store these values and use them at the next simulation step.

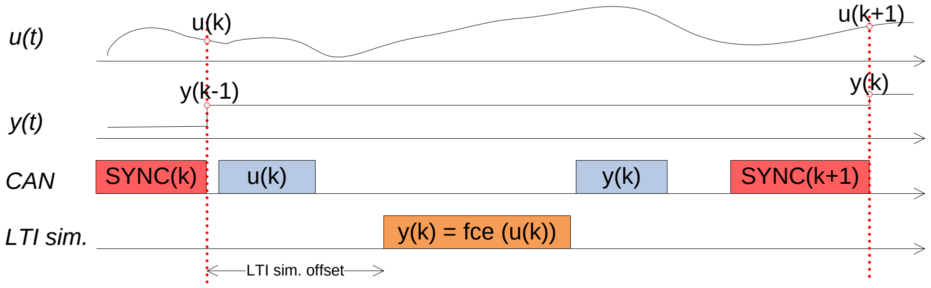


Figure 8 Simulation step Gant chart

The GUI interface exchange data with SimGUI via Ethernet. Model matrixes, scale constants, mapping of signals to sensors and actuators and other configuration parameters are received from SimGUI before simulation start. All measured and computed data are periodically transmitted to SimGUI during simulation. The reference vector can be received from SimGUI before the simulation start or during the simulation.

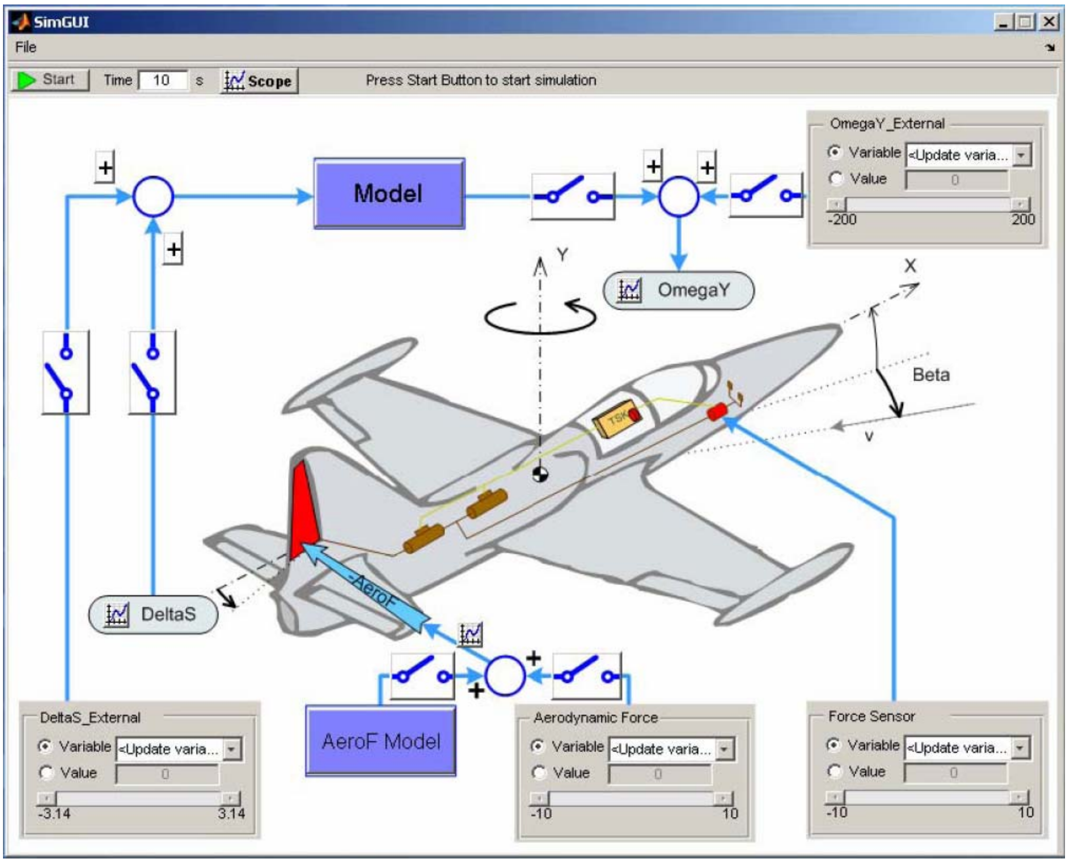
G. SimGUI

The SimGUI is a SW implemented in the Matlab environment. It provides graphical user interface for simulation configuration and data acquisition and visualization. Since it is integrated in the Matlab environment, simulation data can be simply exported to the Matlab workspace for post-simulation analysis.

The main screen of SimGUI is depicted in Figure 9. It provides switch buttons for configuration of signals interconnection, buttons for opening scope windows with predefined signals to be visualized during simulation, buttons opening dialogs for specification plane model matrixes and aerodynamic force model constants and dialogs for reference signals. Reference signal can be either a time sequence specified as a vector defined in Matlab workspace or a constant value that can be changed from the keyboard or via the slider.

When user starts the simulation, SimGUI sends the signal configuration and models to the SimTarget. During the simulation SimGUI periodically reads and visualizes the data received from SimTarget. When user changes a reference value, it is send immediately to the SimTarget. It allows manual control of the simulator.

Except the data from SimTarget, SimGUI receive also diagnostic data from YDC Ctrl via serial line with AMOS protocol. These data can also be selected for visualization during the simulation.



Time 10 s

Press Start Button to start simulation

OmegaY_External
● Variable <Update varia...
○ Value 0
-200 200

DeltaS_External
● Variable <Update varia...
○ Value 0
-3.14 3.14

Aerodynamic Force
● Variable <Update varia...
○ Value 0
-10 10

Force Sensor
● Variable <Update varia...
○ Value 0
-10 10

Figure 9 Main screen of SimGUI

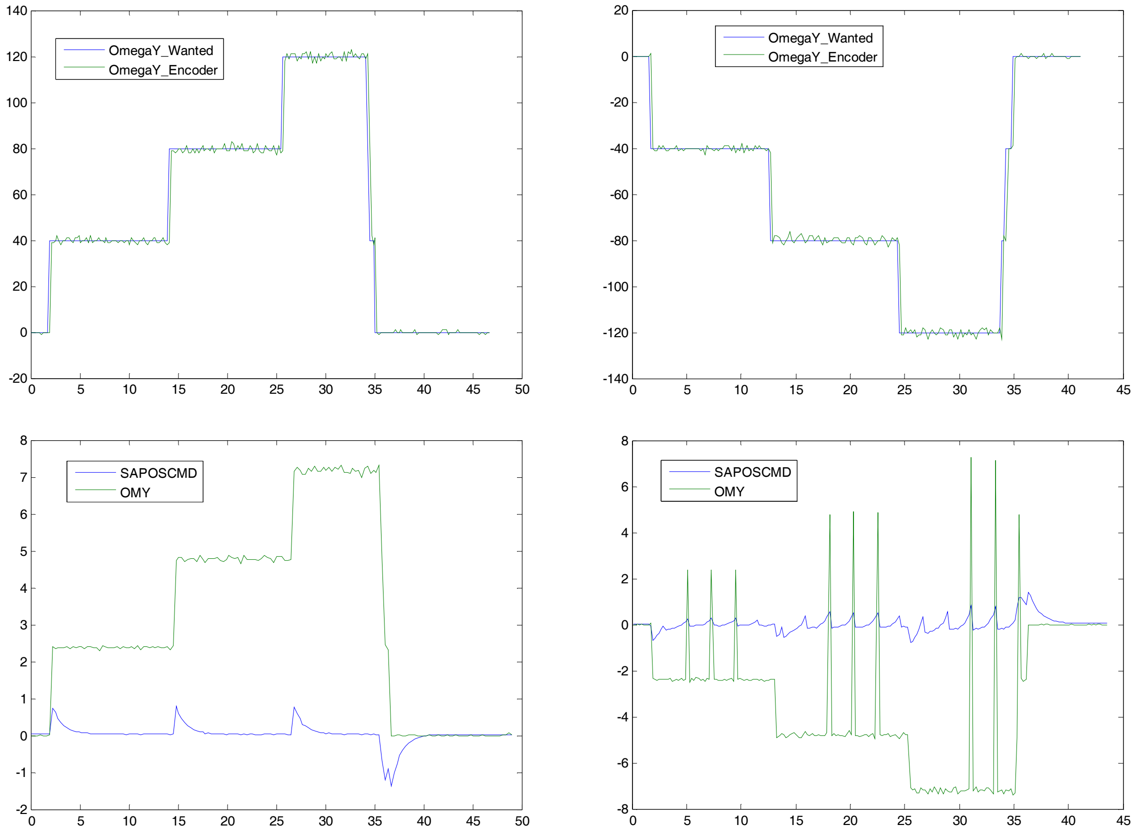
IV. Example of Validation Test

In this chapter we present a simple example of a validation test proving an error in the YDC Ctrl SW. During the fist testes of YDC on the stand we have recognized serial servo oscillations. After analysis of this suspicious behavior in an open loop simulation we have found undesirable peaks in the yaw rate measured by FOG, even though the signal from the encoder of the revolving platform motor has been smooth.

In the conducted experiment we have observed the YDC behavior for different yaw rates. The measured signals are depicted in Figure 10. There is reference value of the yaw rate (OmegaY_Wanted), the yaw rate measured by encoder (OmegaY_Encoder), signal controlling the position of serial servo (SAPOSCMD) and the yaw rate measured by FOG (OMY).

It is clear from Figure 10 that the undesirable behavior occurs only for negative values of the yaw rate. The detail analysis of the OMY signal in the case of negative rotation shows that the undesirable pulses occur at random time, each pulse is always only one sample and the value of this pulse is just equal to the right signal but with opposite sign.

These conclusions help us locate the error very fast. It was caused by mistake in negative sign interpretation in implementation of the protocol used for communication with FOG. Since this communication is asynchronous to control loop of the YDC Ctrl, this error appears only in race conditions. It would be much more harder to find this error without HIL simulator.



a) Rotation in positive direction

b) Rotation in negative direction

Figure 10 Measured data

V. Conclusion

We have described the HIL simulator designed for validation of YDC behavior. The most important components of the simulator has been described and their crucial parameters has been mentioned. The design of the simulator is based on YDC construction, its parameters and requirements on validation tests.

The most important features of the simulator are its open architecture allowing extensibility and integration to the Matlab environment allowing efficient test signal creation and post simulation analysis.

The practical experiences with the developed simulator has been demonstrated on a simple example of a validation test allowing to prove and locate an error in SW of the YDC.

Acknowledgement

This work was supported by Ministry of Industry and Trade of the Czech Republic under Project FT—TA3/044.

References

[1] LabVIEW FPGA in Hardware-in-the-Loop Simulation Applications, National Instruments Corporation, 2003
[2] Jan Slavík, Zadání zkoušky pro zkušební zařízení systému směrového řízení, Rept. AERO Vodochody a.s., 2006