Cicaré CH7-B Engine Governor

Development of an Engine Governor for a Light Helicopter

Zumarraga, Augusto José
Investigación y Desarrollo
CRUX Sistemas Inteligentes
La Plata, Argentina
azumarraga@crux-si.com.ar

Garaventta, Guillermo Norberto
Grupo CAE
INIFTA, Facultad de Ciencias Exactas, UNLP
La Plata, Argentina
garaventa@inifta.unlp.edu.ar

Abstract—**In this work the development of a engine's governor for a light single seat helicopter is presented. Architecture, control loop, hardware and firmware conceptual design is addressed and some difficulties are commented.**

Keywords-component; governor; helicopter; control loop; digital filter; microcontroller; PID

I. INTRODUCTION

Helicopters are the most versatile aircrafts but also the most difficult to fly. The attitude control of a helicopter requires rotation actions on three orthogonal axes simultaneously with a coordinated control of the lift force generated by the rotor (collective pitch) and the engine speed.

The engine speed for a helicopter is the equivalent to the flight velocity in a conventional aircraft. It must be held in a range ±2.5% of its nominal value for safety reasons. If it falls too much, the maximum lift force attainable with the maximum collective pitch angle would not be enough to hold the vehicle's altitude.

In helicopters with manual engine control, the failure of the pilot to hold the rotor velocity above the safety limit is a common cause of accidents. In stationary flight at low altitude, the inertia of the rotor-motor chain brings little possibility to correct the condition before crashing to the ground.

Figure 1. Cicaré CH7-B helicopter

Hence in many countries an automatic control of this parameter is required by the aeronautical regulations.

The present work was undertaken to provide the Cicaré CH7-B helicopter with this capability, permitting the entry of this product in Europe and Oceania markets. The project was constrained with a short development time and with the requirement to match the performance of similar products in the market introducing the minimum complexity to the vehicle.

II. DESING DEVELOPMENT

A. Architecture

This engine governor is a single loop feedback control system. It measures the engine´s regime and acts over its throttles to hold the nominal regime under any action of torque perturbations produced by the collective and tail rotor pitch angle changes.

Although the main controller task is that of a regulator one, a reference following capability was added to provide a controlled startup.

The controller switches to the active state when the engine regime exceeds 80%. This happens by the manual action of the pilot over the throttle handle. Then a reference ramp is internally generated to drive the engine to the nominal speed. The reference is fed to a pre-filter before entering in the control loop in order to avoid overshoot.

The governor automatically disengages when the speed falls below the activation limit. Although the controller prevents the velocity to fall, this occurs normally at land by a manual action of the pilot when he wants to put the machine in an idle condition, since he has control authority over the

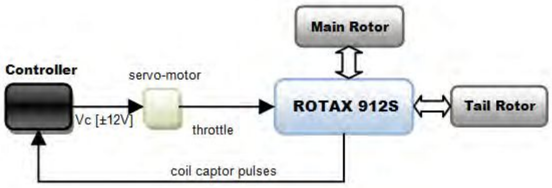


Figure 2. Governor architecture

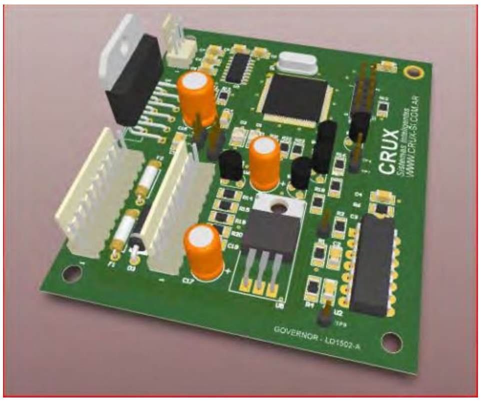


Figure 3. Governor's electronics board

Figure 4. Governor's servo-motor and clutch

governor.

B. Hardware

The governor electronics is implemented in a single board integrating an ATMEL ATxmega64A1 microcontroller, signal conditioning circuits and power electronics (Figure 3). This microcontroller was chosen by its rich set of peripherals, and the efficency of its instruction set for compiling C code.

The primary output is a ±12VCC voltage use to feed a brushed DC electrical servo-motor. This energy is taken from the unregulated battery bus and driven with an H MOSFET bridge commanded by the µC (microcontroller) with a PWM (pulse width modulation) signal and a discrete sense signal.

For security reasons, the servo-motor is coupled to the engine´s throttles commanding mechanism through a friction clutch, giving the pilot the aforementioned control authority (Figure 4).

The primary input is a pulse train generated by conditioning the voltage peaks induced in a coil mounted in the engine´s block by a permanent magnet mounted in the engine shaft. This is used for engine's speed sensing. A Hall Effect sensor is used to measure the rotor speed, but this and other signals are included for status monitoring only. There are also discrete inputs and outputs for the pilot interface.

Rotation speeds are calculated with the time between pulse flanges. The pulses generated by the sensor's coil (with voltages in the order of 100V) are conditioned to CMOS levels before introducing them in the microcontroller. This is the signal used normally by the engine's dashboard tachometer.

The coil sensor output is the voltage transient induced by de rotating magnet on the shaft. This transient is the result of the changes in the gap between the magnet and the coil due to the rotation of the shaft, but has also some contribution of mechanical vibrations. This voltage transient is fed to an integrated circuit with an analog comparator, which triggers a pulse when the voltage crosses a dynamically established threshold. The transients in the air gap produces a voltage transient profile not completely repetitive for every revolution of the shaft, so the precise time of the pulse generation will have variations. The final effect is a perceptible jitter in the period between pulses flanges entering in the µC, producing a significant measurement noise that imposes has to be addressed in the design of the control loop.

The prototype of this controller was implemented with a development kit, which includes an SDRAM memory used to log flight data. An RS232 interface was exposed for data acquisition and commands using a standard industrial protocol. This COM interface was kept in the final prototype for diagnostics and reconfiguration.

The servo-motor, the clutch and the electronics box are the only modules added to the helicopter by the governor, since the magnet and sensing coil is part of the standard engine´s equipment. This minimalist approach was taken to minimize the assembly complexity (this helicopter is sold as a kit) and lowering the probability of failures.

C. Control Loop

1) Plant Identification

In this case the plant dynamics is non-linear, and many dynamical effects change significantly with changes in collective pitch angle. However a linear approach was shown to be satisfactory, mostly due to the fact that under normal operation the changes in the state are small, and only become significant during startup.

From a conceptual analysis we assumed the engine rotation masses, the gear box and the rotational rotor's inertia as a rigid body for the frequency range of interest. The rotor is coupled with the gear box through a clutch, which decouples the rotor when the engine speed is below the rotor speed. This permits the helicopter to glide in the event of an engine stop, but is not important under normal operating conditions.

Since the engine has a natural velocity feedback, at this point we could assume that a first order dynamics should be representative, but this was not consistent with the experimental results.

The identification of the plant dynamics was undertaken through response measurements under different values of collective pitch angle to step variations in the throttle (Figure 5). These measurements where adjusted with the following model:

$$G(s) = \frac{T_z s + 1}{T_p s + 1} \cdot \frac{{\omega_n}^2}{s^2 + 2\xi\omega_n s + {\omega_n}^2}$$

, with the parameters:

$$T_z = 3.6 \quad , \quad T_p = 0.6 \quad , \quad \xi = 1 \quad , \quad \omega_n = 0.29$$

The gain could not be established due to some instrumentation difficulties in the measuring of the throttle and collective pitch commands angles. Other methods, like pseudo-random excitation, could not be applied because of logistics difficulties and lack of time. Although its precariousness, the results become useful to gather a conceptual understanding of the plant dynamics and its bandwidth.

The response to step collective pitch perturbations was also measured and adjusted with a similar model (Figure 6). The observed second order dominant dynamics can be explained by aeroelastic coupling between the rotor flapping motion and its shaft torque. Some preliminary mathematical models of this effects yield consistent results with structural test data and the observed responses. These issues will be addressed in future work.

From those results a 0.07Hz bandwidth was calculated; in order to define the compensator requirements.

2) Loop Compensation and Signal Processing

Indeed the control loop needs integral action to reject the low frequency torque variations necessary to trim the helicopter in different flight conditions. But the integral action is implicit in the servo-motor operation, since from the controller's perspective its dynamics is just an integrator in the feedback path from the perturbation.

There are important friction nonlinearities in the gearbox of this servo-motor. This produced annoying limit cycles in the loop, which were quite difficult to remove. A careful choice of the servo-motor gear ratio and the addition of a bias voltage in the control action reduced this effect up to acceptable levels.

With the minimalist instrumentation approach there is no way to introduce feed-forward action to anticipate the strong effects of torque perturbations during maneuvers; nor to keep the friction nonlinearities out of the loop dynamics. This leads us to pursue a high closed loop bandwidth; and for that we have to introduce a high open loop gain with some derivative action to compensate its adverse effects.

The main drawback is that the measurement noise with a PD controller produces high noise levels in the control action. This does not disturb the engine speed but stresses the servo-motor unnecessarily, so it was necessary to include filtering in the feedback path.

The sampling clock for the PD compensator is an internally generated 10Hz timer overflow interrupt; but for the filter the same speed sensing pulse train is used as sampling event. Indeed this filter sampling rate is not constant, changing the frequency response of the filter with the speed of the engine. However, since in normal conditions this variation is small and the filter bandwidth was kept high enough to maintain it non-dominant in the close loop response, this has no significant impact.

There is another issue with this scheme that must be analyzed. If measurement noise is generated by engine vibrations as we have assumed, the fundamental frequency component of this noise would be the rotation frequency of the engine, which results twice the Nyquist frequency for the filter sampling rate. However, since we have perfect synchronism with this harmonic component, it is aliased to cero frequency and is not being noticed in our measurement. There are also high order harmonic (the R912 is a four stroke, four cylinders engine), but they are in phase with the previous one. We assume that other sources of noise are random in nature, and for that reason aliasing is not observed in our measurements.

A four order Butterworth digital filter was selected. We have to implement it by two second order sections to avoid numerical stability problems experienced with the direct form.

$$H(z) = \prod_{k=1}^{L} H_k(z) = \prod_{k=1}^{L} \frac{{b_0}_k + {b_1}_k z^{-1} + {b_2}_k z^{-2}}{{a_0}_k + {a_1}_k z^{-1} + {a_2}_k z^{-2}}$$

The second order sections are coded in direct form II:

$$v(n) = x(n) - a_1 v(n-1) - a_2 v(n-2)$$

$$y(n) = b_0 v(n) + b_1 v(n-1) + b_2 v(n-1)$$

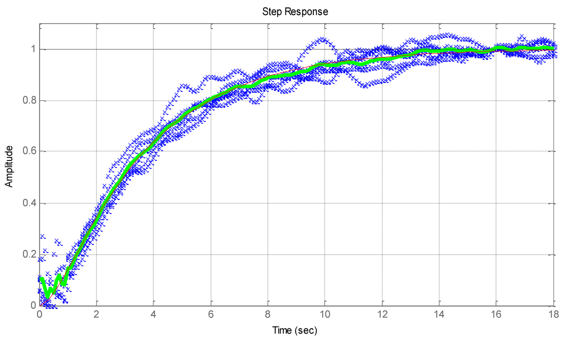


Figure 5. Throttle step response for various collective pitch angles, step response of the identified model.

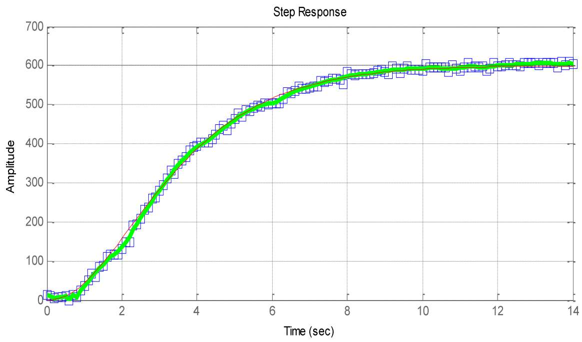


Figure 6. Collective pitch step responses, and step response of the identified model.

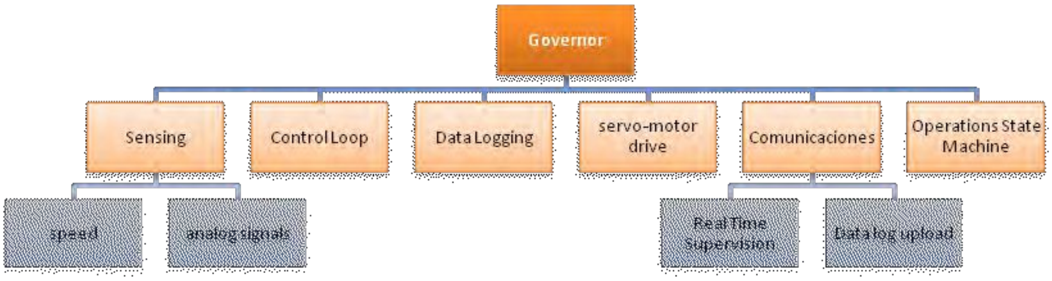
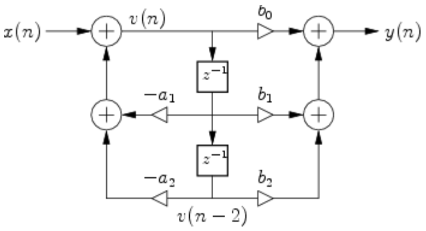


Figure 7. Firmware Function Breakdown Structure



A special code was included for overflow control with extreme input values. These conditions are not expected to happen, but we prefer not to take the risk.

D. *Firmware*

The firmware is implemented in C++ and compiled with GCC with the ATMELs AVR Studio IDE. The code is composed by a reusable core framework and the application specific code.

The core framework provides a hardware abstraction layer, so the µC hardware specific details are hidden in the application code. This framework is mainly a collection of class and function templates; and since with C++ templates most of the code's polymorphism is resolved in compilation time, this does not introduce any significant processing overhead in runtime, but allows a strong object oriented and reusable design.

The application code is based in the functional breakdown structure shown in Figure 7. Each main function is encapsulated in a singleton class, with minimal dependencies with the others.

Data flow for A/D converters and UART buffers is managed by DMA (Direct Memory Access) without intervention of the CPU. This is one of the more remarkable features of the XMEGA 8bit processor architecture. The frequency measurements are performed with capture timers and their interrupts. The filtering calculations are performed in these interrupt handlers.

The behavioral features are executed by a state machine driven by a 10Hz timer interrupt. This timed event triggers the PD control loop when the running state is the "active mode".

A switch placed behind the throttle handle, at the front of the collective lever forces the state machine to enter the

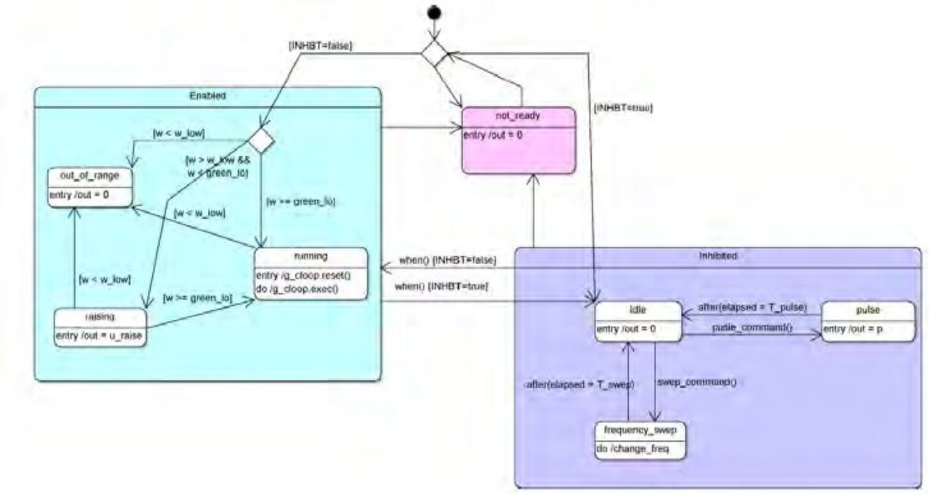
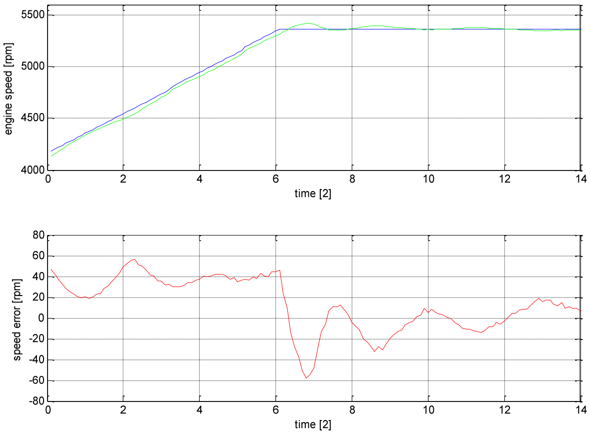


Figure 6. Firmware state machine

inhibited state, disabling any governor action on the throttle. In this state the governor can be commanded to execute some special diagnostic functions, like generating user defined control action pulses. This activity is controlled through the COM interface.

III. CONCLUSIONS AND FUTURE WORK

At the present the first versions of this governor is in production, and its performance satisfies the helicopter manufacturer's initial expectations. The following figure shows the loop response in the *rising* and *running* state.



However we think it could be better. Particularly we need to decrease perturbations sensitivities.

Further experimental work would allow us to correctly identify the plan dynamics, in order to refine the loop compensator; but we do not think we can enhance the loop performance by a compensator tuning, due to the strong restrictions on bandwidth already mentioned.

Our strategy is the inclusion of perturbation feed-forward action. This can be done by mechanical coupling between collective pitch control mechanism and throttle command, as was implemented by the helicopter manufacturer, but implies more mechanical complexity. For that reason some instrumentation work is being undertaken to provide measurements of the collective pitch, and handle this electronically. Complexity does not seem to be a problem with this approach since we only need to add an incremental encoder, but the additional cost must be taken into account. The XMEGA architecture includes an "event system" which permits the interaction between several hardware modules without intervention of the CPU. As a special feature this system includes "quadrature decoders" that can be connected with any I/O pin and used to feed a timer/counter to track the encoder position completely by hardware. With a second timer we could measure velocity also.

With respect to the overall functionality of the product, the capabilities of the ATMEL XMEGA microcontroller allow a lot of features to be included, particularly for engine supervision and history recording. This is other point under evaluation, and some decisions need to be taken related to de convenience of expand the governor's functionality, or instead introduce an independent electronic module for that.

ACKNOWLEDGMENT

This work was done with the collaboration of Daniel Hamann, a member of ours electronic engineering team who was responsible for the PCB design.

We want to thanks also to Marcos Knoblauch for the review of this paper, Gaston Bonnet and Cristian Bottero for providing structural test data for the rotor; and Fernando Cicare, CEO of Cicare S.A., who was our official test pilot.

REFERENCES

[1] J.G.Proakis, D.G.Manolakis, "Digital Signal Processing - Principles, Algorithms and Applications", 1996, Prentice Hall.

[2] A.V. Oppenheim, R.W.Chafer, J.R.Buck "Discrete-Time Signal Processing", 1999, Prentice Hall.

[3] M.H.Hayes, "Statistical Digital Signal Processing and Modeling", 1996, John Willey & Sons, Inc.

[4] V.K.Madisetti, D.B.Williams, "The Digital Signal Processing Handbook", 1999.

[5] Engineering Staff of Analog Devices, DSP Division, "Using the ADSP-2100 Family Volume 1", 1999, Prentice Hall.

[6] B.Stroustrup, "The C++ Programming Language", 2000, Addison-Wesley

[7] Bruce Eckel, "Thinking C++", January 2000, Planet PDF.

[8] ATMEL Corporation, "XMEGA A MANUAL", http://www.atmel.com/avr