Configurable Digital Fractional Order Hold Device for Hybrid Control Systems

KOLDO BASTERRETXEA AND RAFAEL BÁRCENA
Department of Electronics and Telecommunications
University of the Basque Country (UPV/EHU)
IITUE Bilbao. La Casilla plaza, 3, Bilbao
SPAIN

Abstract: - It is proved that special hold devices such as the fractional order hold device (FROH) can improve, if properly tuned, the performance of hybrid control systems. Although the applicability of these devices has been theoretically studied in numerous works, there is no report of real experimentation with digital controllers that include special hold devices. This lack of real application experiments is mainly consequence of the difficulties encountered in implementing such a hold device with the processing speed and configurability that real-time digital controllers require. This paper describes a totally configurable digital design of a FROH device, adequate for any hybrid control system platform. The implementation of an eight bit resolution FROH has been performed and tested on a field programmable gate array (FPGA).

Key-Words: - Hold device, FROH, configurable hardware, FPGA, digital controller, D/A conversion

1 Introduction
In feedback control systems, the positions of the plant zeros notably bias the achievable performance of operation –see, for instance, [1] and its references-. In the last years –see, [2], [3] and cited references-, various research works have been focused on the behavior of discrete ceros of digitally controlled systems as a function of used signal reconstruction method and sample time. In this context, it has been demonstrated that using properly adjusted fractional order hold devices (FROH) for control signal reconstruction, a more stable position of discrete ceros can be achieved compared to systems that use the more common zero order hold (ZOH) or first order hold (FOH) devices [4]. In [4] an analytic method is proposed to obtaining the optimum value of the FROH parameter for each discretization. Moreover, it is demonstrated as well that there is no limitation in the sample period for the application of the above mentioned tuning method [5].

While the use of FROH and other advanced hold devices has been studied at theoretical level and experimentation can be performed in simulation environments, there are no real experimentation reports, as far as we know, of hybrid control systems with embedded special hold devices. This is very likely due to the difficulty in achieving an adequate hardware implementation of such a device, which must be highly configurable to meet different digital control requisites (sampling periods, signal resolution etc), and fast enough to make device delay negligible compared to the used sampling period.

In this paper we present the design and implementation of a fully configurable digital FROH device with adjustable gain parameter and fast operation. Implementation has been performed on a FPGA which assures the configurability of the device for different control requirements, so the obtained device is suitable for any real-time hybrid control system.

2 Hybrid Control System with FROH-based Signal Reconstruction
The design of the FROH device described here has been faced considering the constraints imposed by our actual real time digital controller experimentation system in what refers to sampling frequencies, clock signals and word-length. Still, the goal has been to obtain a universal design, easily adaptable to any digital control platform, so a parameterized hardware description has been performed. In this sense, the use of reconfigurable hardware as target technology is of base importance to assure the required programmability of the device and to allow the necessary modifications in order to adapt the device to the requirements of different controllers. In what follows we briefly describe our hybrid control experimentation platform and its main characteristics.
Fig. 1 shows a general view of the real-time control experimentation system based on PC and used as reference for this design. The design and tuning of the digital controllers is made under the Mathworks programming environment at the host PC. The implementation of the computation of the control algorithms is performed by the Real Time Workshop, and the xPCTarget package is used to execute the algorithms on a real-time core running in the Target PC. The target PC, together with a multifunction data acquisition card (DAQ), constitute the embedded real-time digital controller which performs the various control strategies to be experimented over different plants. The FROH device must be placed between the digital output of the DAQ and the DAC/signal conditioning circuitry that feeds the plant’s signal input.

The DAQ actually attached to the Target PC is a NI-PCI6024E by National Instruments. Its main characteristics are:
- 8 TTL digital inputs/outputs.
- Two timing outputs with 24 bit resolution.
- 16 analog inputs with 12 bit resolution.
- 2 analog outputs with 12 bit resolution.
- 200 Ks/sec sampling frequency.
- 20 MHz internal reference clocking.

Accordingly, the output signals of the DAQ, which constitute the input signals of the hold device, are the following:
- Digital control signal (UkT): 8 bit positive normalized value, UkT ∈ [0,1).
- Clock signal (clk): 1 MHz 50% duty cycle clock obtained from DAQ timing output 1.
- Triggering signal (trigger): triggering pulse signal generated every sampling time and synchronized with the UkT output.

The three signals are synchronized over the timing base signal of the DAQ. The choice of a 1 MHz frequency clock signal obeys to the restriction imposed by the DAQ timing output, since above this frequency it produces a poor signal shape, inadequate for clocking. Fig. 2 shows the signaling scheme between the DAQ and the hold system.

It must be emphasized that the delay introduced by the hold system, included D/A conversion and the signal conditioning stage, must be “sufficiently small” compared to the sampling period (T) used by the controller. This means that the maximum delay must be between one and two orders of magnitude below T to assure that the control tuning remains valid for the system. In the following section we describe in detail the design of the FROH device.

3 Digital Circuit for Fractional Order Hold

A FROH is an adjustable signal hold device for hybrid control systems. Its main advantage over the ZOH and the FOH, as stated before, is its adjustable gain parameter β, which provides a certain freedom degree for the placement of the zeros of the discretized plant and, therefore, may improve the performance of certain feedback digitally controlled systems, particularly when the sampling period T is relatively high compared to the plant dynamics. It is a hold device with an adjustable linear hold function...
and one element of memory. The hold function can be analytically described by the following expression:

\[
u(t) = u(kT) + \beta \left[ \frac{u(kT) - u((k-1)T)}{T} \right] (t - kT)\]

for \(kT \leq t \leq (k+1)T\)

Where \(T\) is the sampling period, \(k\) represents the \(k^{th}\) input sample, and \(\beta\) is the adjustable gain parameter.

If a digital implementation is made, we obtain a hold function approximated by successive steps (Fig 3). We call this hold device the approximated fractional order hold or AFROH [7].

![Fig. 3 Hold function of a FROH and a two step AFROH approximation.](image)

Being \(N\) the number of steps to approximating the hold linear function, we can write:

\[
u(t) = \left( 1 + \frac{2l-1}{2N} \right) \beta u(kT) - \left( \frac{2l-1}{2N} \right) \beta u((k-1)T)\]

\(k + \frac{l-1}{N} \leq t \leq (k + \frac{l}{N})T, \quad l = 1, ..., N\)

It is obvious that, the more steps the device is able to produce, the better and smoother approximation to the continuous ideal function is obtained.

### 3.1 Digital design of the hold device

Various constraints must be considered when designing a digital AFROH device. First, the gain factor \(\beta\) of the device must be adjustable. The tuning of \(\beta\) is made off-line and is calculated to obtain an optimum behavior of the controlled system. In addition, we want a selectable value of \(N\) within a minimum value \(N=2\) and a maximum value \(N_{max}\), which will be limited by the resolution of the system and its maximum computing speed or clock frequency. The resolution or digital word-length \((res)\) of the device will be considered as a design parameter also, as it can vary for different hardware platforms (our word-length in this design is 8 bits). Beside this, it must be considered that the sampling period \(T\) may change for different control applications, so this is an element of the vector \(P\) of design parameters too: \(P = [\beta, \text{res}, N, T]\). Let us fix the limits of the parameter values, more precisely, of the \(T/N\) rate: considering that the clock frequency is limited to 1MHz, we will take a minimum value of the sampling period of \(T_{min}=0.1\text{ms}\), so we will limit the value of \(N_{max}\) to 64 steps (powers of two are used as it will be controlled by a binary counter). This way, the amount of clock cycles required for each step is \(T/T_{clk}N_{max}\), which is in the minimum bound because, as we will see, the hold device has a latency of one clock cycle. In case the minimum sampling period would need to be reduced an order of magnitude or so (that is to \(T = 0.01\text{ms}\)), \(N_{max}\) should be limited to 8, which could be adequate for the most demanding plant dynamics.

The circuit has been designed following the block scheme depicted in Fig. 4. It can be observed that, even if this design would require two clock signals, all sequential elements are clocked by the same signal of period \(T_{clk}\), conforming a unique clock domain. This way we assure the total synchrony of the system and simplify static timing analysis. To perform the selective timing of the various modules a signal controller block \((\text{counter/timing})\) in Fig.4), which produces adequately timed control and enable signals, has been designed. The circuit operates internally with double word-length (16 bits) as the block \(\text{step}\) (Fig.4) encloses an internal multiplier to calculate the value to be added to each approximating step in order to produce the next one. This block operates with signed values, allowing the use of negative values of \(\beta\) to generate both negative and positive slopes.

Two additional elements have been attached to the circuit to prevent the output from exceeding the variation range of the control signal which, with an 8 bit word-length, is \([0, 0.99609375]\). One is an AND gate placed at the output of the output register, which is fed with the sign bit of the register content and zeroes the output when this bit changes to ‘1’ (negative value). The other one is a comparator that checks the output value and disables the output register when the maximum limit value is reached. This enable signal is activated again when a new control value is provided at the input \((\text{trigger} = ‘1’\)). In fact it is very unlikely that an underflow or an overflow happen because the optimum values of \(\beta\) are usually negative, so the hold function slope make the signal tend to its steady state value.
Let us analyze in detail the operation of the blocks step and counter/timing:

**Counter/timing:** The input parameters of this block are $T$ and $N$ ($T_{clk}$ is supposed to be known as well). It encloses two pulse counters, one for the interval of period $T$, and other one for the interval of period $T/N$. Combined with a comparator-based logic, it provides three timing signals essentials for the correct operation of the device:

- The signal $ctrl$: This is the input selection signal of the multiplexer. During the first hold interval (first step) this signal is asserted to ‘1’, selecting the system input signal $U_{kT}$, which is directly routed to the output register. During the rest of the sampling period this signal is asserted to ‘0’, and the multiplexer transmits the output signal from the adder.
- The signal $sum$: This signal is asserted each time a new step must be generated. Combined with the $overflow$ signal, it controls the output register’s enable signal. When active, the register content is updated at each clock pulse, adding it to the value provided by the step block and so generating the next step.
- The signal $reg$: This signal is activated two clock cycles before a new control value $U_{kT}$ is provided to the device. It enables the input register of the step block to capture the “previous” value of the control signal $U_{(k-1)T}$ to calculate the new step amplitude.

**Step:** The input parameters of this block are $N$ and $\beta$. It is an arithmetic block that calculates the signal amplitude that must be added to each step in order to generate the next one. The FROH is a hold device with one memory element, since the previous value of $U_{kT}$ must be registered to compute the hold function. Thus, this block performs the operation $[U_{(kT)}-U_{((k-1)T)}]/\beta/N$. Considering $\beta/N$ as a configuration parameter rather than as a function variable, this block only encloses a substractor and an internal multiplier that multiplies the difference between successive samples by a constant value.

### 3.2 Circuit description, synthesis and simulation

A register transfer logic (RTL) VHDL description of the hold device has been written based on the scheme presented in Fig. 4. The word-length ($res$), $T$, $N$, and $\beta$ complete the set of parameters of a parametric hardware description file, assuring the validity of the code under any variation of the vector $P$ and making the device suitable for any controller. We used Xilinx ISE9.1i for code debugging and circuit simulation. The synthesis has been made with the XST (Xilinx Synthesis Tool), and placement and routing processes on an Spartan-3E family FPGA by the Xilinx ISE implementation tools. Fig. 5 shows circuit operation for a sampling period $T = 0.1\text{ms}$, a number of steps $N = 32$, and $\beta = -0.8$.

The circuit latency is one clock cycle, imposed by the output register, as the input is routed directly to this register. This way, for a $T_{clk} = 1\mu s$, we assure that the delay introduced by the hold device is “sufficiently small” compared to $T_{min}$. Obviously this delay can be reduced by augmenting the clock frequency (if possible).
The target device has been a Xilinx 3S500efg320-4 FPGA. The synthesis and timing details after placement and routing are listed below:

**Macro Statistics**
- # Multipliers : 1
- 9x9-bit multiplier : 1
- # Adders/subtractors : 4
- 16-bit adder : 1
- 9-bit adder : 2
- 9-bit subtractor : 1
- # Registers : 4
- 16-bit register : 1
- 9-bit register : 3
- # Comparators : 1
- 9-bit comp. lessequal : 1

**Device utilization summary:**
Selected Device : 3s500efg320-4
- # Slices: 49 out of 4656 1%
- # Slice Flip Flops: 42 out of 9312 0%
- # 4 input LUTs: 88 out of 9312 0%
- # IOs: 19
- # bonded IOBs: 19 out of 232 8%
- # MULT18X18SIOs: 1 out of 20 5%
- # GCLKs: 1 out of 24 4%

Total equivalent gate count: 1.026

**Timing Summary:**
Speed Grade: -4
Offset in : best case Achievable 12.118ns
Offset out : best case Achievable 11.648ns
Clk minimum period: 14.170ns

The figures show that maximum clock frequency for this implementation is 73.914 MHz, which would permit the use of controller sampling periods of 0.01ms and less, and to generate more than 64 steps at each sampling period if necessary.

4 D/A conversion

Once the AFROH generates the digital values to approximating the hold function, a D/A conversion must be performed. The conversion scheme to be applied is conditioned by the operation frequency and the maximum delay requirements. The resistor ladder based parallel D/A conversion is the fastest and the most suitable for high frequencies, but an analog circuit is required and its accuracy may be affected by noise and temperature changes. A more compact option that can be embedded inside the FPGA together with the AFROH, is a pulse-duration based conversion scheme. In addition, pulse modulated signals are suitable to directly drive many electric machines and are widely used in motion control. The simplest option is a Pulse Wide Modulation (PWM) circuit followed by a passive first order low-pass filter, but PWM-based D/A conversion have some drawbacks. First, since the frequency of the output bitstream is the same as the input rate, its bandwidth is relatively wide and large time constants are needed for the output filter. Second, the noise from pulse generation tends to concentrate at this frequency, so it is not filtered at the output. Both of these problems are reduced with a sigma-delta modulator [8].

Hence, a sigma-delta modulator has been embedded and connected to the AFROH output (Fig. 6). However, keeping the resolution of the system would require a very high clock frequency. In our case, even considering the restrictions imposed to the design, the worst case ($T_{min} = 0.1ms$ and $N = 64$), would require 256 clock pulses each 1μs to keep a 8-bit resolution modulation, and that means a 256 MHz clock frequency. To implement a first prototype we have reduced the D/A resolution to 5 bits. That means producing a 32 MHz clock signal, and this is precisely the maximum frequency we can obtain from a Spartan-3E DCM (Digital Clock Manager)

---

**Fig. 5** Simulation output of the AFROH device for $T=0.1ms$, $N=32$, and $\beta=-0.8$. The waveform diagram shows the moment in which a new control value $U(kT)$ is sampled.
frequency synthesizer output when a 1MHz clock is applied to its input. This resolution reduction requires scaling down the input, but this only implies a hardwired bit shifting. The simulation of the whole system operation is depicted in Fig. 7.

5 Conclusion
Implementation of special hold devices such as the FROH for real-time digital control has always been problematic because of required operation speed and device configurability. Nowadays, the spread and cost reduction of high performance reconfigurable hardware provide the means to face the design and implementation of such a device for real-time hardware-in-the-loop experimentation with advanced digital controllers. The configurability of FPGAs allows for a parametric design that assures the adaptability of the hold devices to the controller changing requirements: control signal resolution, sampling period, parameter tuning of the hold device, and accuracy of the hold function approximation. The implementation of the FROH described here is a first step to achieve an advanced digital controller experimentation bench with special hold devices, which we hope will allow us to go deeper into this attractive research field.

References: