An Adaptive System for Robust Analog Signal Sample/Hold

Chunyan Wang

Concordia University

Abstract - **In this paper, a scheme of adaptive system producing a signal sample that is equal to the input is proposed. The system consists of a simple signal sample generator and some logic gates forming a feedback loop. The sample generator is modeled as a multi-dimensional nonlinear circuit and the digital code is applied to determine the coordinates of its operating point. By using the negative feedback loop, the code is update step by step to get the right operating point for the system to produce the sample that is the closest to the input signal. This updating process makes good use of the non-linearity of the system to have the step size variable in such a way that it is large at the beginning of the process to have a quick convergence and small at the end for a small residue error. Because the signal sample is recorded by a digital code and it can be reproduced easily without problems occurring in analog signal storage and reproduction, the system can be used for a robust analog signal sample/hold. As an example of implementing such an adaptive system, a current-sample/hold circuit has been designed and presented in the paper.**

keywords - **Adaptive system, analog signal sample and hold, convergence rate, nonlinear circuit, residue error, variable step size, successive approximation.**

I. Introduction

An adaptive system can automatically update its own parameters, according to the signals it receives, so that its output is made to adapt to the inputs. Such a system can be very sophisticated to perform complex functions. It can also be made very simple, for low-cost and easy applications, and effective for not-complex-but-difficult operations. Analog signal sample/hold is one of such operations.

Analog signal sample/hold is one of the most important operations in mixed signal processing and data conversion systems. The quality of the sample/hold has impact on the performance of the systems. Traditionally, analog switches and capacitors are used to sample a signal and to hold it for a limited period [1]. The quality of such a sample/hold operation, e.g. the precision of the signal sampling and the duration of the sample held with the same precision, is very limited by the switching noise and the charge leakage of the capacitors.

.Chunyan Wang, Department of Electrical and Computer Engineering, Concordia University, 1455 Maisonneuve West, Montreal Canada H3G 1M8, Email: chunyan@ece.concordia.ca.

To avoid the problems related to analog devices, one may consider to use digital approaches. To this end, we can first use an A/D converter to record an analog signal by a digital code so that it can be stored digitally and then use a D/A converter to reproduce the analog signal. Such a sample/hold circuit may be complex, in terms of its structure, and its performance metrics are usually determined by those of the D/A convertor and the signal comparators employed. Some research has been done to achieve a high precision of a sample/hold operation without a high-precision D/A conversion. One of such designs is to use a sub-binary radix D/A converter to reduce the gap between 2 successive numbers [2], which improves the resolution, while reducing the dynamic range, if no extra D/A conversion bits are added. A good precision can also be obtained with low-cost D/A converters. In [3], a resolution of 14 bits is achieved by using a structure involving two 8-bit D/A converters that may not have a great linearity but provide all together 16 bits to makes the gap smaller. It is therefore common that the precision of the operation is improved by reducing the gap, i.e. the step size in general term, which requires, in fact, a larger number of bits of the signal applied to the D/A circuit.

Sampling and holing an analog signal by means of digital codes is no doubt very effective without problems related to switching noise and leakage. However, in this type of circuit, a better precision is related to a larger number of bits, which usually leads to a more complex circuit and longer time of the sampling process. In this paper, a scheme of very simple adaptive system is proposed, aiming at a simplicity of circuit structure and fast tuning process with a reasonable precision, and the design of a current sample/hold circuit is described. The simulation results of this circuit are also presented.

II. Scheme of the Adaptive System for Signal Sample/Hold

The proposed scheme is based on a negative feedback loop that is widely used in many adaptive systems and successive approximation circuits. The basic frame is shown in Fig. 1. The objective of the operation in this scheme is to generate a discrete-time signal $g(n)$ and to make it equal, or very close, to $s(n)$, the input signal to be sampled. The signal sample generation is controlled by a digital code $\Delta(n)$ that is updated by the error signal $e(n)$. As long as $e(n)$ is non-zero, $\Delta(n)$ tends to be updated to make $g(n)$ closer to $s(n)$. For a given circuit block of the signal

sample generation, the signal *g(n)* can, in fact, be presented by the digital code Δ*(n)*. In other words, the analog signal *g(n)* can be reproduced by the code Δ*(n)*, which is similar to a DAC circuit reproducing the same analog signal with the same input code.

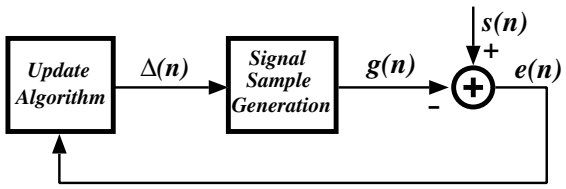

Fig. 1. General scheme of the adaptive system. To sample the input signal *s(n)*, we assume that *s(n)* maintains a constant level during the sampling process that may last a number of clock cycles.

As the scheme shown in Fig. 1 is a negative feedback system, it attempts to make *g(n)* converge to a given *s(n)*. The following issues should be addressed in the design of the system.

- Convergence. If the system starts with an arbitrary value of *g(n)*, it is desirable that *g(n)* approaches *s(n)* after a very short period of time. The convergence is related to the step size of the signal updating. A larger step size leads to a quicker convergence but usually resulting in a larger residue error.
- Precision. When *g(n)* is updated to be the level closest to *s(n)*, the difference, i.e., *e(n)* = *s(n)* - *g(n)*, should be small enough with respect to the level of *s(n)*. A good precision results from a small step size, which may lead to a slow convergence, as mentioned above. Moreover, if the dynamic range of the input signal is wide, a large number of bits for Δ*(n)* will be needed.
- Facility of circuit implementation regarding the convergence time and precision. The number of clock cycles needed to reach the convergence is related to the update algorithm and the number of bits of Δ*(n),* while the duration of the clock cycle depends on the circuit complexity. Also, if one intends to use a large number of bits to improve the precision, the improvement can effectively be done only if the analog blocks used for signal sample generation and the signal comparison have the required wide dynamic range and precision.

To implement the general scheme shown in Fig. 1, the challenge would arise if one attempts to use a small number of bits, for a simplicity of the circuit and fast convergence, to achieve a wide input dynamic range and reasonable precision of the sample/hold operation. One can succeed with such an attempt by using the bits in an "economic" manner. The number of bits of Δ*(n)* is related to the maximum number of the steps in the process of updating *g(n)*. The step size has to be made variable to minimize the total number of steps. More precisely, the step size should vary in such a way: (i) larger steps in the beginning of the process and smaller ones when the convergence is about to reach, and (ii) larger steps when *s(n)* has a large magnitude and smaller ones if *s(n)* is smaller. To make such a variation, the system presented in Fig. 1 needs to have the following two characters.

1. The digital code Δ*(n)* is of "floating-point" type. Like a code of floating-point number having some bits for "exponent" and the other for "significand", the code of Δ*(n)* is partitioned into at least two groups, one to define the scale of the signal to be generated and the other to specify the signal level within the range defined by the effective scale.
2. The signal generation, controlled by Δ*(n)* is made to have non-linear characteristics to provide a variable gain.

A functional block, such as one generating a signal sample, involves some parameters determining its characteristics. For instance, in the case of a MOSFET used as a current generator, the current i_D is determined by the gate-to-source voltage v_{GS} and the drain-to-source voltage v_{DS}. This current generator can be modeled as a multi-dimensional system, in which a given set of coordinates (v_{GS}, v_{DS}) specifies a sample of i_D. Moreover, in a particular operation mode, e.g. the saturation mode, the current i_D is more sensitive to v_{GS} than to v_{DS}, i.e. $\frac{\partial i_D}{\partial v_{GS}} > \frac{\partial i_D}{\partial v_{DS}}$. The nonlinear behavior in each dimension is different from that in any of the other dimensions.

In general, a nonlinear signal generator can be characterized as $g(n) = f\,[p_1(n), p_2(n),\ldots p_N(n)]$, in which $p_1, p_2,\ldots p_N$ are the parameters determining the signal level. It can be modeled as a multi-dimensional system of which $(p_1(n), p_2(n),\ldots p_N(n))$ is the set of coordinates specifying the operating point of the sample generation at a given time. By updating the coordinates the signal sample is updated. As mentioned previously, the bits of the digital code Δ*(n)* are partitioned into groups and each of them is applied to update one of the coordinates. If $\frac{\partial g}{\partial p_1} > \frac{\partial g}{\partial p_2} \ldots > \frac{\partial g}{\partial p_N}$, *g(n)* will be changed at a higher rate by updating p_1 than any of p_2 to p_N. Thus, by incrementing or decrementing p_1, *g(n)* is updated with the largest step size whereas the finest step can be implemented by means of p_N. The negative feedback loop shown in Fig. 1 is made to set up, by iterations, all the coordinates, starting with the most sensitive one, i.e. p_1, and ending with p_N to make the finest steps while *g(n)* is converging to *s(n)*. In this converging process, the digital code Δ*(n)*, with its bits grouped, is being "trained" to specify the coordinates of the operating point for the signal sample generator to produce a copy of the input *s(n)*. Therefore, the code Δ*(n)* represents the coordinates, rather than a digital number, in the multi-dimensional system generating the signal.

With the principle described above, the general scheme shown in Fig. 1 can be easily implemented in a circuit structure as one example shown in Fig. 2. Assume that the signal sample generation is determined by three parameters p_1, p_2 and p_3. The code $\Delta(n)$ is grouped into $\Delta_{p1}(n)$, $\Delta_{p2}(n)$, and $\Delta_{p3}(n)$, each of which is used to set up one of the three parameters. The update algorithm can be made very simple, providing an increment or decrement of each of the bit-groups of $\Delta(n)$. One can use, for example, three unipolar counters, each for up-counting or down-counting. The process for $g(n)$ to converge to $s(n)$ should be expected as shown in Fig. 3.

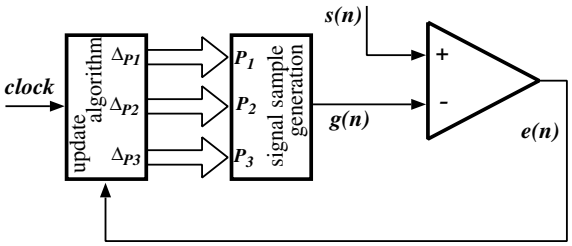

Fig. 2. Structure for the sample/hold operation. The block of the update algorithm produces a digital code $\Delta_p(n)$ consisting of three sub-codes, namely $\Delta_{p1}(n)$, $\Delta_{p2}(n)$, and $\Delta_{p3}(n)$. Each of them is applied to set up one of the parameters p_1, p_2 and p_3 of the block for the signal sample generation in order to make $g(n)$ to converge to $s(n)$ with different step sizes.

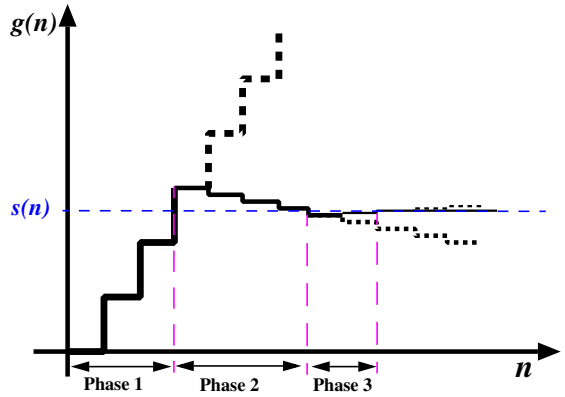

Fig. 3. Process of $g(n)$ converging to a given $s(n)$ in the structure shown in Fig. 2. The X-axis is the time presented as the number of clock cycles. The solid line represents $g(n)$ over the time with the thickest one in the first phase and the thinnest one in the last phase. During Phase 1, $g(n)$ progresses with a large step size while the parameter p_1, to which $g(n)$ is very sensitive. is being set up by an incrementing Δ_{p1}. Phase 1 ends when $e(n) = s(n) - g(n)$ changes its sign from positive to negative, which latches Δ_{p1} and starts Phase 2 with Δ_{p2} updating. Right after this moment, $g(n)$ changes its course. Instead of following the rest of the Phase 1 track, indicated by the dashed thick line, it starts to step down at a smaller pace. When $e(n)$ again changes its sign, Phase 3 will start and $g(n)$ is then steps up at the finest pace. The following change of the sign of $e(n)$ marks the end of the process.

The convergence process shown in Fig. 3 has successive phases to make $g(n)$ approach $s(n)$, with a coarse update with large steps during the first phase and the finest update during the final phase. If a phase start with $g(n) < s(n)$, like Phase 1, the sub-code $\Delta_{pi}(n)$ will update p_i in such a way that $g(n)$ is made to step up, and the phase will be ended by $g(n) > s(n)$. The end of each phase, is marked by the change of sign of the error signal $e(n)$. During the succeeding phase, $\Delta_{pi+1}(n)$, applied to set-up p_{i+1}, updates itself in the opposite direction, and $g(n)$ is updated also in the opposite direction but with a smaller step size, as $g(n)$ is less sensitive to p_{i+1} than p_i. The signal $g(n)$ is updated with a step size very large at the beginning and very fine at the end to have a fast convergence and a small residue error.

It should be noted that in Fig. 3 if the Y-axis is scaled logarithmically, the step size in a lower level will be smaller than that in a higher level in the same phase, which helps to have a larger range of signal sampling. Another issue to be noticed is that using a low-sensitivity parameter, such as p_3, helps to facilitate fine tuning. In the following Section, a current sample/hold circuit is presented as a design example of applying the proposed scheme. It also shows how the logarithmic scale and the fine tuning can be easily implemented in the circuit.

III. Design of a Current Sample/Hold Circuit

Based on the diagram shown in Fig. 2, a current sample/hold circuit has been designed. In this circuit, The signal sample generation and comparison are implemented in the circuit shown in Fig. 4. The block of the update algorithm consists of three sub-blocks to generate $\Delta_{p1}(n)$, $\Delta_{p2}(n)$, and $\Delta_{p3}(n)$, respectively. Each sub-block involves a shift-register with a 1-bit control to stop shifting asynchronously.

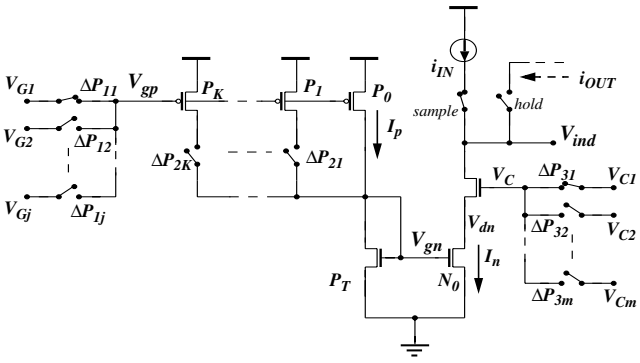

Fig. 4. Circuit diagram of the current sample generator combined with a current comparator. The current sample I_n depends on V_{gn} and V_{dn} that are determined by the state of the switches controlled by the signals ΔP_1, ΔP_2, and ΔP_3.

In the current sample generator shown in Fig. 4, the current sample I_n is compared with the input current i_{IN},

and the voltage V_{ind} indicates the sign of i_{IN} - I_n. If i_{IN} - I_n > 0, V_{ind} tends to be at a high level, otherwise at a low level. The current I_n is determined by V_{gn} and V_{dn} in a nonlinear manner. The former is controlled by the switch signal of ΔP_{11} to ΔP_{1j}, to select the voltage level of V_{gp} for a coarse adjustment of V_{gn}, and that of ΔP_{21} to ΔP_{2K}, to determine the amount of current to be inject at the node V_{gn} for its fine-tuning. The latter, the voltage V_{dn}, is related to V_C determined by the signal of ΔP_{31} to ΔP_{3m}. As $\frac{\partial I_n}{\partial V_{gn}} \gg \frac{\partial I_n}{\partial V_{dn}}$, V_{gn} is first to set up and V_{dn} is then updated, which is carried out in three phases. The circuit can be initialized with $V_{gp} = V_{G1}$, $V_C = V_{C1}$ and all the switches controlled by ΔP_{21} to ΔP_{2K} turned on. Under this initial condition, V_{gp} decrements during Phase 1 and this phase ends when V_{ind} changes from high to low, indicating a change of the sign of i_{IN} - I_n. During Phase 2, the signal ΔP_{21} to ΔP_{2K} turns the switches off one by one, while V_{gn} decrements at very small pace, till the sign of i_{IN} - I_n changes back. In Phase 3, if V_{ind} is detected to be higher than a given level, V_{dn}, the drain voltage of the NMOS N_0 will then be increased, step by step by means of lifting V_C, to fine-tune the transistor current I_n up. The end of this phase is marked by a decrease of V_{ind}.

It should be mentioned that the circuit illustrated in Fig. 4 can also function with an initial condition that $V_{gp} = V_{Gj}$, $V_C = V_{Cm}$ and all the switches controlled by ΔP_{21} to ΔP_{2K} turned off. In this case, the current I_n will converge to the level of i_{IN} after evolving in the opposite direction to that in the previous case.

During the three-phase process described above, the current sample I_n varies in almost the same way as *g(n)* shown in Fig. 3. In each of the phases, I_n is updated in a unipolar manner. If a phase starts with i_{IN} - I_n > 0, such as the case of Phase 1, it ends with i_{IN} - I_n < 0, which also marks the start of the succeeding phase. Each time when the sign of i_{IN} - I_n changes, the level of V_{ind} will be changed to indicate such a state. Thus, V_{ind}, equivalent to *e(n)* in Fig. 1, can be used to control the start/end of the phases, i.e. to enable or not-enable the shift registers in the block of the update algorithm.

In the current sample/hold circuit, if the voltages V_{G1} to V_{Gj} are of linear progression, the step size of updating I_n is exponentially increasing during Phase 1. Thus, the step size in this phase is related to the magnitude of i_{IN}. A lower i_{IN} gets a smaller step size to have a finer resolution. Also, V_{gp} is set up not only to coarse-adjust V_{gn} during Phase 1, but also to determine ΔI, the current flowing in each of the PMOS transistors P_1 to P_K during Phase 2. Moreover, these transistors are sized the same and each has a ΔI equal to a small fracture of I_p. However, the sizes of the PMOS transistors can be made different to update I_n in a non-linear scale for a wide coverage.

It should be noted that at the end of each phase, I_n is "over-updated" by an approximate one-half step. A compensation can be easily done by putting the one-half step back. For instance, one can add the $(K+1)^{th}$ PMOS branch in parallel with all the other PMOS ones shown in Fig. 4, with the gate size of P_{K+1} being one half of that of P_K and the switch P_{K+1} turned on at the end of Phase 2.

It should also be noted that a fine step size of updating can be realized without manipulation in a very fine scale. By applying a binary control to inject/remove ΔI, one can fine-adjust V_{gn} with a step size of millivolts, as the sensitivity of V_{gn} to ΔI is low. Also, in the final phase, I_n is fine-tuned in a nano-Ampere scale by means of updating V_{dn} in a 100-millivolt scale, thanks to the fact that I_n is not very sensitive to V_{dn}. Thus, one can make a good use of the character of low-sensitivity to facilitate a fine-tuning of a signal sample.

The current sample/hold circuit, including the current sample generator shown in Fig. 4, has been simulated with the models of a 0.18 μm CMOS technology. In this circuit, ΔP_1 is a signal of three bits controlling eight switches, which allows V_{gp} to choose one of the eight voltage levels ranged from 1.3 V to 1.58 V, while the supply voltage is 1.8 V. Another set of eight switches is for the second phase. One bit signal of ΔP_3 is to control the two switches in the final phase. Thus, the equivalent number of bits of the digital code controlling the signal sample generation is seven. With the very modest 7-bits, the residue error rate of the current sample generation, defined as $|i_{IN} - I_n|/ i_{IN}$, is below 5% throughout the current sample range from 3.5 nA to 4.5 μA. The waveforms obtained in the simulation with i_{IN} = 66 nA are presented in Fig. 5.

IV. Conclusion

A scheme of adaptive system for robust analog signal sample/hold has been proposed in this paper. This scheme is based on a negative feedback, like many other adaptive systems. The state of the system is updated by means of applying very simple digital code and it evolves in a unipolar manner, i.e. simply incrementing or decrementing, in each operation phase. The nonlinearity of the system is well explored to realize a variable step size in the convergence procedure to achieve a good precision and fast convergence time. Based on this scheme, a current sample/hold circuit has been designed. This circuit consists of simple circuit blocks generating the current sample and its digital control, and the generated sample can converge to the input signal with a small residue error but without a large number of clock cycles.

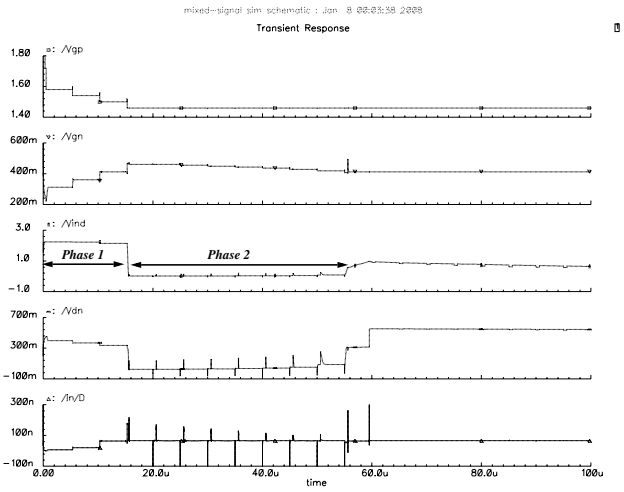

Fig. 5. Simulation waveforms obtained with i_{IN} = 66 nA. The voltage V_{ind} is first at a high level then changes to a low level while V_{gn} is first increasing with large steps and then decreasing with smaller steps. Phase 1 ends when V_{ind} changes to a significantly lower level. The following change of $V_{ind,}$ this time to a higher level, indicates the end of Phase 2. The waveform of the current sample I_n is found at the lowest position in the figure.

REFERENCES

[1] A.A. Arnaud and M.M. R. Miguez, "On the evaluation of the Exact Output of a Switched Continuous Time Filter and Applications," *IEEE Transactions on Circuits and Systems I*, vol. 99, no. 2003, 2008.

[2] W. G. Bliss, C.E. Seaberg, R.L. Geiger, "A very small sub-binary radix DAC for static pseudo-analog high-precision memory," *Proc. IEEE Midwest Symposium on Circuits and Systems*, Aug. 1992, vol. 1, pp. 425 - 428.

[3] G. Scandurra, C. Ciofi, G. Giusi, M. Castano, G. CannatCannata, "Design and Realization of High-Accuracy Static Analog Memories (SAMs) Using Low-Cost DA Converters," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 6, pp. 2275 - 2280, Dec. 2006.