Implementation of DSP algorithms for ultrasonic measurement applications

E. Kazanavičius, R Venteris

Digital Signal Processing Laboratory, KUT

Studentų 50-214c, Kaunas LT3031, LITHUANIA.

Introduction

Modern sonar and radar measurement applications effectively use digital signal processing (DSP) tasks such as signal averaging, filtering, and correlation. For each individual application these tasks should be implemented obeying the application requirements such as signal lengths, required calculation speed, precision. Therefore, recent DSP technologies are helpful due to optimal algorithm and architecture matching. Implementation of signal correlation in many cases is also the subject under research. In sonar applications the correlation of long signal arrays and short computation period is often needed [5]. This problem is actual in applications of sonar tracking moving objects; also sonar vision based control systems for mobile robots where high computation speed is required in order to perform real time control [2], [3], [4].

The paper presents analysis of a smart sonar vision application for mobile robot control and DSP algorithms used for it. Also, there is given the formal description of the application algorithm, DSP architecture and system implementation.

DSP algorithms in sonar distance measurement and object vision methods

Sonar object vision applications use the distance measurement method (Fig. 1) [1], [5], [6]. The reference signal x_{ref} consisting of a certain coded sequence is emitted at some direction by the sonar transmitter at time moment t_0. The propagating signal reflects itself from the environment objects and is accepted by the receiver. The reflection signal y is equal to the sum of x_{ref} with time delay Δt and of the environment noise η_{env}.

$$y(t_0+\Delta t) = k\cdot x_{ref}(t_0)+\eta_{env} \qquad (1)$$

where k - coefficient depending on distance to the object and environment properties.

The distance d to the object is directly proportional to the delay time Δt and described:

$$d = \Delta t\cdot c_{env}, \qquad (2)$$

where c_{env} - coefficient defined by environment properties.

The value of Δt is found by computing the correlation function (CF) of x_{ref} and y and detecting the argument of its maximum value:

$$\Delta t = \arg\max(corr(x_{ref}, y)) \qquad (3)$$

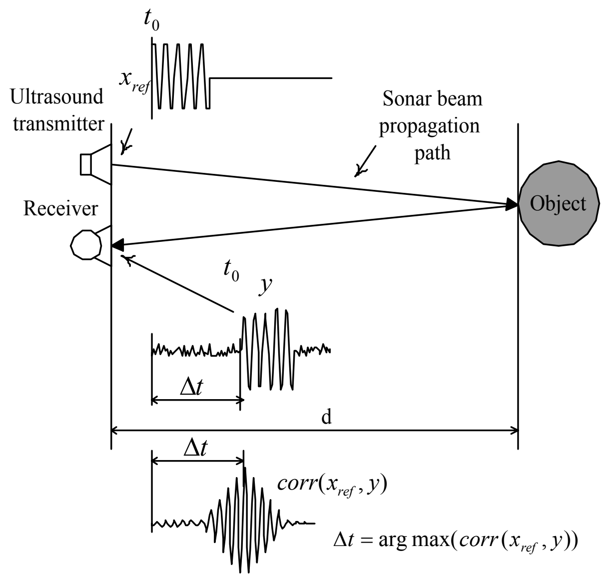

Fig. 1. **Method of sonar distance measurement.**

Smart sonar object vision applications often use several transmitters with orthogonal reference signals and receivers working simultaneously [2], [3], and [4]. Therefore, the formal description for the application with M channels is:

$$\left\{\begin{array}{l} \vec{X}(t) = \begin{bmatrix} x^1_{ref}(t_o) \\ \dots \\ x^M_{ref}(t_o)\end{bmatrix}, \\ \vec{Y}(t) = \begin{bmatrix} y^1(t_0+\tau_1) \\ \dots \\ y^M(t_0+\tau_M)\end{bmatrix} = \begin{bmatrix} k\cdot x^1_{ref}(t_0)+\eta_{env} \\ \dots \\ k\cdot x^M_{ref}(t_0)+\eta_{env}\end{bmatrix}, \\ R^i_{xy}(\tau) = \frac{1}{T}\int\limits_{\tau=0}^{T} x^i_{ref}(t_0)y^i(t_0+\Delta t^i+\tau), i=1..M, \\ \Delta t^i = \arg\max(R^i_{xy}(\tau)) \\ d^i = \Delta t^i\cdot c_{env} \end{array}\right\} \qquad (4)$$

Obviously, the main tasks in the application are:

1) Cross-correlation,

2) Detection of CF maximum value and its argument.

Since the application is considered to be for industrial conditions [4] thus system robustness to environmental noises should be increased. Therefore, long reference signals are used. In this case the correlation becomes an algorithm with long computation period, and some approaches are needed to implement it in real time control systems.

Correlation function algorithms and computing time evaluation

Two algorithms can be used to implement CF of two signals.

1st method: sum-of-products

In this case normalized cross-correlation function r_{xy} of signal *x(N)* and *y(M)* has the following expression:

$$r_{xy}(k) = \frac{1}{M}\sum_{i=0}^{M-1} x(i+k)\cdot y(i)\text{ , } k=0..N-1 \qquad (5)$$

Respectively, auto-correlation function of *x(N)* is expressed as:

$$r_{xx}(k) = \frac{1}{N}\sum_{i=0}^{N-1} x(i+k)\cdot x(i)\text{ , } k=0..N-1$$

The formulas are similar to the FIR filter algorithm. However, if the lengths of signal arrays are tremendous compared with conventional FIR filters then big number of calculations is needed.

2nd method: fast correlation algorithm with Fast Fourier Transform (FFT)

This method is effective to correlate long signal arrays. The expression for fast CF is:

$$r_{xy} = FFT_D^{-1}\left[FFT(x)\cdot FFT^*(y)\right] \qquad (6)$$

where FFT^{-1} is notation for inverse FFT and $FFT^*(y)$ is complex conjugate for $FFT(y)$. The flowchart of fast CF is given below (Fig.2).

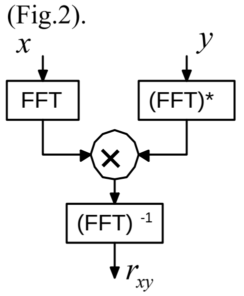

Fig. 2. **The flowchart for fast CF calculation.**

Note, if the reference signal x_{ref} is constant there is no need to calculate its FFT. Therefore, the number of calculations can be reduced by factor 1.5.

Formal comparison of calculation speed in both methods

Assume the computation speed of CF depends on the number of needed operations.

The expression (5) has $M\cdot N$ multiplications and $(M-1)\cdot N$ additions. The number of operations in the 1st method is:

$$m_1 = (M-1)\cdot N(a.) + M\cdot N(m.) \approx 2\cdot M\cdot N \qquad (7)$$

For analysis of the 2nd method the data from [1, p.77] on complex additions and multiplications number in FFT algorithm was taken. Consequently, the number of operations for fast CF with constant x_{ref} FFT array:

$$\begin{aligned} m_2 &= 2\cdot N_{FFT} = \\ &= 2\cdot(2N\log_2 N(m.) + 3N\log_2 N(a.)) = \\ &= 10N\log_2 N \end{aligned} \qquad (8)$$

The ratio $K = \frac{m_1}{m_2}$ formaly is the ratio of computation speed in both correlation algorithms which depends on variables N and M - y and x_{ref} lengths. Substituting real values for M and N, the obtained value can be the criterion to choose realisation method. Several K values are shown in the table below. There are K values for M=N (second column), and for constant N=8192 with variable M (third column).

Table 1.

Length of reference signal array, M	N=M $K = \frac{m_1}{m_2}$	N=8192 $K = \frac{m_1}{m_2}$
64	-	1
128	-	2
512	11.38	7.9
1024	20.48	15.8
2048	37.24	31.6
4096	68.27	63.2
8192	126.03	126.03

Notice, that for small M values sum-of-products algorithm becomes more effective than fast correlation algorithm with FFTs.

Realisation would be different for general purpose DSPs and programmable logic (FPGA). DSPs execute code sequentialy and the given formal comparison is much valid for them. FPGA platform enables realise parallel architectures therefore, often it is more simple and effective to implement parallel sum-of-products architecture rather than the complex architecture of FFT processor.

However, considering large signal arrays and using general purpose DSP processors the fast CF algorithm can be sufficiently effective for correlation.

Application analysis and the DSP architecture

Distance measurement method was applied for an object vision system performing control of the automated robot. The vision system consists of M identical distance measurement channels operating for different space directions. After measuring distances in all M channels the collected distances are used to reconstruct the environment model and manage robot movement.

Referring to (4) and assuming fast correlation algorithm, the formal application algorithm is:

GIVEN:

$\bar{x}^m = [x_0^m, x_1^m, ..., x_L^m], m = 1..M$, *//reference signal*

$\bar{y}^m = [y_0^m, y_1^m, ..., y_L^m], m = 1..M$ *//input signal*

INIT: *//initialization stage*

For *m=1* ***to*** *M channels* ***do parallel***

[

For *k=0* ***to*** *(L-1)* *//X= DFT(x)*

$$X_k^m = \sum_{n=0}^{L-1} e^{-2\pi jkn/L} x_n^m$$

]

end do parallel

Start=0;

BODY: *//body of algorithm*

For *m=1* ***to*** *M channels* ***do parallel***

[*Start=1;****]***

[

While *Start=0* ***do;***

$generate(\bar{x}^m)$;

]

[

While *Start=0* ***do;*** *//Wait for Start*

For *k=0* ***to*** *(L-1)* ***do*** *//input m signals*

$y_k^m = ADC^m$;

For *k=0* ***to*** *(L-1)* ***do*** *//compute Y=DFT(y)*

$$Y_k^m = \sum_{n=0}^{L-1} e^{-2\pi jkn/L} y_n^m;$$

For *k=0* ***to*** *(L-1)* ***do*** *//multiply X and Y*

$RC_k^m = X_k^m \cdot Y_k^m$;

For *k=0* ***to*** *(L-1)* ***do*** *//calculateIDFT(RC)*

$$R_{xy}^m(t) = \sum_{n=0}^{L-1} e^{-2\pi jkn/L} \cdot RC_k^m;$$

$d^m = \arg\max(R_{xy}^m)$; *//calculate distances*

]

end do parallel

//surrounding media reconstruction algorithm from distance data

$\vec{S} = F_{HOST}(d^m)$;

//send results to supervisory system

$out(\vec{S})$;

END.

Given model is the subject for DSP architecture synthesis. Architecture would be optimally implemented if the principle *"application–algorithm–scheduling–architecture"* is completely satisfied. Following this principle the highest level DSP architecture for the application was suggested (Fig.3).

Further architecture decomposition would enable to formalize internal architectures of the blocks based on FPGA platform.

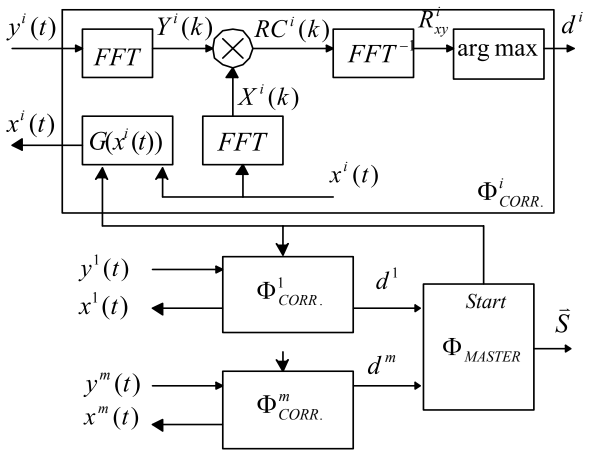

Fig. 3. **Highest architectural level for the application.**

Architecture implementation

The DSP architecture was implemented as a multiprocessor DSP system (Fig. 4).

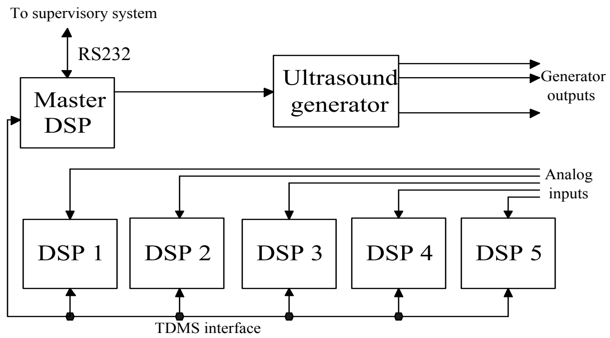

Fig.4 **Architecture of implemented DSP system**

The system consists of 6 DSP modules. Five DSPs correspond to Φ_{corr}^M function, where number of channels M=5. These modules perform parallel signal acquisition, compute R_{xy}^M and retrieve distances d^M from it. The Φ_{corr}^M modules are controled by the Master DSP which performs Φ_{MASTER} function.

The modules have a unique architecture and are realised on Texas Instruments DSP TMS320C50 platform operating at 80MHz. Figure 5 shows the internal architecture of DSP1-DSP5 modules. Each of DSP1-DSP5 computes the fast CF of 8192 sample input signal and optional reference signal The computing time for 8192 point fast CF is 110 miliseconds.

The master DSP has the same hardware architecture but different software. All 6 modules are connencted to the net via 10Mbit time division multiplexed serial interface. The master DSP performs ultrasound generator control, DSP1-DSP5 control, distance data collecting, surrounding media reconstruction and transmission of the results to a supervisory system via RS232 interface.

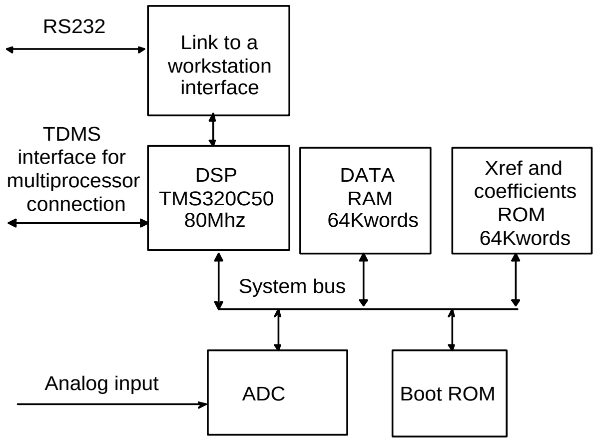

Fig. 5**. Architecture of the DSP module.**

Conclusions

Main disadvantage of the implemented DSP system is rather small its flexibility. The system based on traditional DSPs has the architecture which optimally matches the application algorithm only at the highest decomposition level. At the lower architectural levels DSP modules have fixed architecture which is not effectively exploitable and on the other hand sets some constraints, e.g. fixed wordlenght, impossibility to perform operations parallel, etc. In perspective such DSP applications would be effectively realised using programmable logic technologies (FPGA) thus improving system parameters such as lower cost and energy consumption, smaller dimensions, higher operation speed and shorter implementation period.

Acknowledgements

This project was carried out as a part of INCO-COPERNICUS project No. ERB IC15-CT96-0726.

References

1. **Kažys R., Kundrotas K., Dzimidavičius V., Mažeika L., Borkovski A.** Programmable ultrasonic range finder for mobile robot.– Robotersysteme, Springer-Verlag, pp. 101-106, 1991.
2. **Kažys R.** Smart ultrasonic sensor for semi-autonomous robots. – Mechatronics'98, Elsevier Science Ltd., pp. 489-494, 1998.
3. **Kažys R.** Smart systems for robot vision. – MMAR'98, pp. 827-832, 1998.
4. **Kažys R.** Smart ultrasonic system for mobile robots. – IAPR-MVA'98, Fujitsu Makuhari Lab., pp. 347-350, 1998.
5. **Ifeachor E.C., Jervis B.W.** Digital Signal Processing: A Practical Approach. - Addison-Wesley, pp. 71, 185-250, 1993.
6. **Denbigh Ph.** System Analysis & Signal Processing. – Addison-Wesley, pp.438-473, 1998.
7. **Fathi E.T., Krieger M.** Multiple processor systems: What, Why and When. – IEEE Computer, pp.23-32,1983.

E. Kazanavičius, R Venteris

Signalų apdorojimo algoritmų taikymas ultragarsinių matavimų sistemoms

Reziumė

Ultragarsinio ir radarinio matavimo taikomuosiuose uždaviniuose naudojami įvairūs skaitmeninio signalų apdorojimo (SA) algoritmai: signalų vidurkinimas, filtravimas, koreliacija ir t.t. Kiekvieno taikomojo uždavinio algoritmų realizacijai reikalinga speciali SA sistemos architektūra, kuri turi būti sudaryta atsižvelgiant į uždavinio keliamus reikalavimus (signalų masyvų dydžius, maksimalią leistiną algoritmo vykdymo trukmę, skaičiavimų tikslumą ir pan.). Signalų koreliacijos algoritmo realizacijos problema aktuali ultragarsiniuose matavimuose, sekant judančius objektus. Minėtuose uždaviniuose koreliuojami dideli signalų masyvai, tačiau skaičiavimai turi neilgai trukti, kad sekimo sistema funkcionuotų realiu laiku. Straipsnyje pateikiama objektų regos uždavinio, paremto ultragarsinio atstumo matavimo metodu, ir jam realizuoti reikalingų SA algoritmų analizė. Uždavinys aprašomas algoritmu, iš kurio pagal realizacijos principą "uždavinys - algoritmas – architektūra" sintezuojama SA architektūra. Pabaigoje pateikiama architektūros realizacija ir išvados.

DOI: 10.5755/j01.u.32.2.7936