DSP Implementation Of Interpolative Type Controllers

Adrian Korodi
Department of Automation and Industrial Informatics,
"Politehnica" University of Timisoara, Timisoara, Bd. Vasile Parvan 2,
RO-1900, Romania,
e-mail: akorodi@aut.utt.ro

Lucian Peana
Department of Automation and Industrial Informatics,
"Politehnica" University of Timisoara, Timisoara, Bd. Vasile Parvan 2,
RO-1900, Romania,
e-mail: lpeana@aut.utt.ro

Abstract. **This paper presents two similar methods to implement an interpolative type controller by using a Digital Signal Processor Starter Kit. First structure is a 2-dimensional interpolative controller with static behavior and the second one is a 1, 2, or 3-dimensional interpolative controller with dynamic behavior. At the end of the implementation was conceived a graphical user interface for each interpolative type controller. The graphical user interface help's the programmer and also the user to reconfigure the controller. The results are satisfactory, and in most cases meet and also improve the control system requirement.**

1. INTRODUCTION

It's well known that fuzzy and neural-type methods are implementations of the approximate reasoning, as well as that approximate reasoning implementation is in fact interpolation (Zadeh, 2002). The idea of using interpolator type blocks was thoroughly considered a method based on the idea of transposing control rules in support points of a control hyperspace (Dragomir, 2001).

Recently published papers shows that the digital signal processor (DSP) represent a viable alternative to implement controller's witch require high computation speed like neural network (Kim, 2001), some nonlinear controller's (Youn, 2002; Benchaib, 2003). The implemented controller's are very efficient due to the high-speed computation (Vukosavic, 2003).

The main objective of this paper is to attain an interpolative type controller (IC), in order to emphasize the possibility of improving the control system performances, using a signal processor (TMS320C31) and also to assign it a graphical user interface with the easy to use purpose.

2. INTERPOLATIVE TYPE CONTROLLERS

The interpolator controllers represent a control alternative, which in certain control schemes can replace the controllers obtained through known designing methods with tables and interpolation algorithms, and in most cases by modifying the tables the control system will lead to a better results. In this context we have to discuss some aspects regarding the implementation of IC and tuning of interpolative table.

The IC's are based on the implementation of control algorithms through support points implanted in interpolative blocks, which are used either as such, or integrated in dynamic structures. The interpolative blocks are, in essence, interpolative tables, which contain a finite number of support points (in fact the tables contain support values that, together with the corresponding labels define a support point).

Once it is established the control law to obtain a first structure of the IC, it will be proceeds to extract some support points of control table related to the control law. After this operation the table will replace in the control system structure the initial controller and will implement a "copy" of the initial system. The next step in the development of interpolator control is the correction phase, dedicated to improve the behavior of the control system via table modification.

The *planning correction*, beside creating the table, assumes the following:

- *Selecting of an improvement criterion*, that may refer to the local or global aspects of the behavior of the control system during relevant situations.
- *Selecting the correction* aims to find in the table some domains, which will be submitted to modifications. In this context, the correction method may consist in modifying the values of the table or in modifying granularity of the support points.

The modification of the support values, the content of interpolative table is done point by point and only in some points. The points whose support values are modified are chosen from the selected domains on the basis of an "anticipating thinking".

The *modification of granularity* is associated with modification of the dimensions of the interpolative table. The growth of the volume of the table must be controlled from point of view of the DSP disposal memory.

3. IMPLEMENTATION STRATEGIES

A. DSK minimal description

The TMS320C31-50 one of the fastest floating point signal processor (40ns instruction cycle time), produced by Texas Instruments, offers the possibility of processing 25 MIPS and 50 MFLOPS.

The *DSK (DSP Starter Kit)* module is a kit realized around this signal processor. It is made by hard and soft components (TLC32040 analog interface circuit, windows-oriented debugger, simplifying code development and debugging capabilities, etc.) building a flexible support for implementation of the IC's. The DSK has the advantage of a relatively low price and also the access to all the signal processor's pins.

B. Implementation aspects

The implementation of an IC on the DSK module had to be realized reducing as much as possible the execution time but also minimizing the necessary memory space. The program obtained must have a good processing time as much as to be well made. For example: it is obvious that, by using tables with raised granulation (tables with many points), the interpolative implementation leads to great results, but as much as the interpolative table is bigger (the performance is growing), the occupied memory space is bigger and also the search in the table requires more time. A good result can be obtained using tables with limited granulation too (tables with few points). The technological progress into the domain of the computing devices is now getting away this impediment, in the sense that any application will dispose of the amount of memory required in order to use tables that reach the necessary granulation. From the speed's point of view, the TMS 320C31 signal processor helps to obtain a good execution time and the DSK module gives the possibility to add a new external memory that can solve the space problem. Also the DSK's assembly language, used in the implementation process, has, in comparison with C a great advantage: fastness.

Except the memory limitation a second embarrassment led to the necessity of using a multiplexer (MAX-307) because the DSK board has only one analogical input and both situations needed more than one external analogical input signal (2-in the first structure and 1, 2, 3-in the second one). The MAX 307 multiplexer helps to selecting and taking over the input signals.

C. Program structure

The main points of the implementation program are:

o *Signal acquisition:* The signal processor take-up the analog input signal through the analog-digital converter (ADC) TLC32040, transpose it in real number representation and the obtained value is drive to the search procedure.

o *Searching subroutine:* use the transpose value to find two points from the support place between witch the value is positioned. The algorithm *extract from the table* the value located "around" the place indicated by the input values.

o *The interpolation algorithm:* represents the last function to be accomplish. Having the search results, the interpolation module extract the support values corresponding to the identified labels, and implement the interpolation formula.

D. Presentation of some modules

The IC use two type of data: support place values and supporting values of interpolative table. In theory this values are simply represented together like a table with the heading elements being the label values (see fig. 1). In our case is more effective to store all the support place values like vectors at the beginning of memory (L_I+L_J+L_K support place values):

```
PlaceI   .float    -1.3, -1.1, -0.8, … , 1.1
PlaceJ   .float    -1.3, -1.2, -0.5, … , 1.2
PlaceK   .float    -1.3, -1.1, -0.1, … , 1
```

and after these the support values of the table (L_I*L_J*L_K values):

```
table    .float    -1, -1, … , -0.166, … , 1.1, 1.3
```

The low-level assembly language do not allow the programmer to use matrix data for access the elements using more than one indexes, so the support points table are stored like a vector and the access to an element is done with an indexed address obtained by the following formula (for 3-dimensional table).

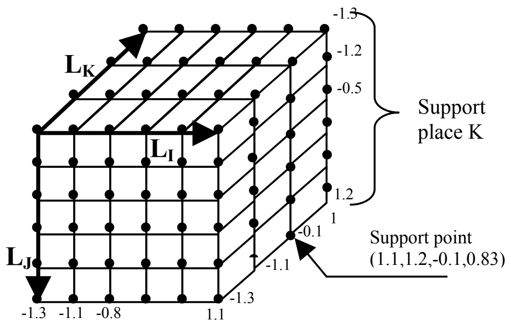


Fig. 1. Interpolative table

$$L_I \cdot L_J \cdot (K-1) + L_I \cdot (J-1) + I \qquad (1)$$

where I, J, K represents the indexes of specified dimensions axis and L_I, L_J, L_K are the table dimensions.

o The *signal acquisition*. According to the dimension of the IC (with 1, 2 or 3 dimensional table) the DSP is connected to the multiplexer MAX-307 (see fig. 2). In the first structure (two-dimensional table), the TMS320C31 processor's pin, XF1 was connected to the MAX 307's selection pin, A0, and it was programmed to take over one of the two input signals each time. For the second structure, the TMS320C31 processor's pin, XF1 and the timer 1's output pin, T1, were connected to the multiplexer (at A1 and A0 selection pins). The number of the external input signals is not known and therefore the module is conceived to be able to select and take over 1, 2 or 3 signals (depending on the settings made in the program).

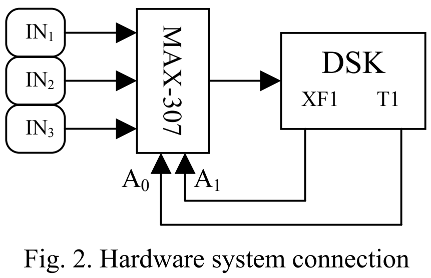


Fig. 2. Hardware system connection

The “signal acquisition” function is made according to the analog digital 14-bits converter (ADC). The analog input (±1.5V) is automatically converted by the ADC into a signed integer value (V_{SI}). The integer values V_{SI} from the ADC is transformed in a real value (V_R) used by the search subroutine. The real values are calculated by the formula:

$$V_R = 1.5(Volts)\cdot\frac{V_{SI}}{2^{13}} = 0.0001831\cdot V_{SI} \qquad (2)$$

o The *search subroutine* consists in scanning the support place vectors Vn,p (with $n=\overline{1;3}$ and $p=\overline{1;p_n}$ and Vn,p<Vn,p+1) in order to find a greater value than the input signal value represented internally as V_R.

$V_{n,1}$	$V_{n,2}$	…	$V_{n,in}$	…	V_{n,p_n}

Fig. 3. Support place vector

In order to prevent the overflow situation the last value Vn,pn of each support places is set to be greater that any possible input value V_R. The *search algorithm* (fig 4)perform the same operation for each input signal IN_n converted internally into a real value V_R (V_R=GET IN_n).

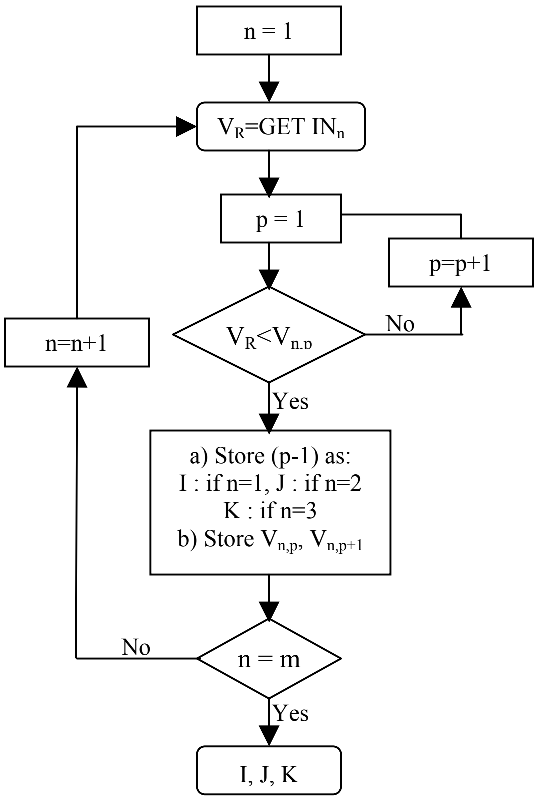


Fig. 4. Search algorithm

At the and of the search subroutine the indexes I, J, K will point to the essential support value.

o In the following it will be detailed the *extracting support points algorithms*, for the 3-dimensional tables. The 3-dimensional interpolation procedure requires 8 support points for interpolation calculus, but the indexes I, J, K (from the search subroutine) match only to one point, which in fact is the corner of the same cube (fig. 5).

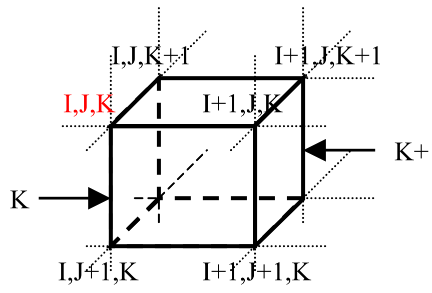


Fig. 5. The 8 support points

By starting from the point I, J, K it is very easy to find the other points, by using the index registers (IR0, IR1). To extract all the support values, three steps are required, showed as follows:

1. Extracting (I,J,K) and (I+1,J,K) $\rightarrow$ one dimension
2. Extracting <1>, (I,J+1,K) and (I+1,J+1,K) $\rightarrow$ 2–dimensional table,
3. Extracting <2>, (I,J,K+1), (I+1,J,K+1), (I,J+1,K) and (I+1,J+1,K+1) $\rightarrow$ 3-dimensional table. (3)

where (I, J, K) represents the support value located in position I, J, K of the table.

The interpolation module implements the output calculation by the specified way and the indicated formulas. In both structures it is used the *linear interpolation* method. For the calculation it will be used 2, 4 or 8 support points (depending on the table dimension).

o The resulting of the interpolation is a real value witch have to be send to the digital-analog converter (DAC). The real value are converted internally with the formula (4) in an integer value that can be used by DAC.

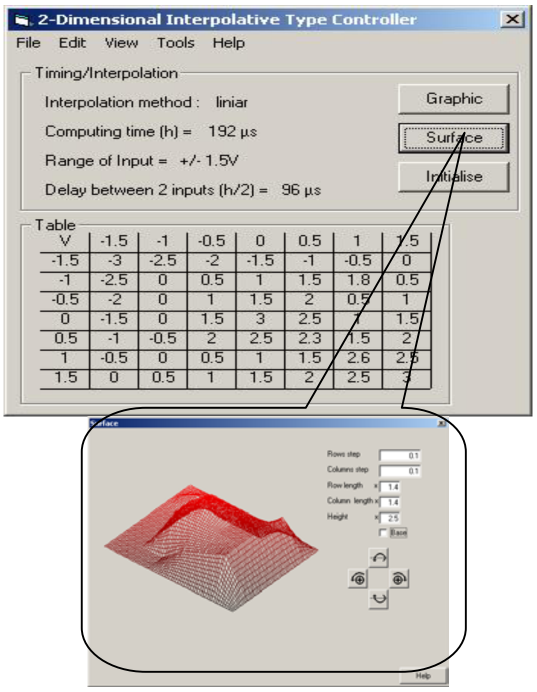
$$V_{SI} = 2^{13}\cdot\frac{V_R}{3(Volts)} = 2730.66\cdot V_R \qquad (4)$$

The value V_{SI} is calculated for the range ±3V of the analog output signal.

E. *The first structure*

The first structure implements a 2-dimensional IC which respects all the steps from sections D. The IC has a static behavior, all the blocks that can offer the dynamical behavior must be added externally.

In some cases even for the programmer is difficult to change the IC settings by modifying the assembly code, disadvantage eliminated by designing a graphical user interface (see fig. 6), which allows to change almost all the parameters of the IC and helps to have a better view of the table hyperspace.


2-Dimensional Interpolative Type Controller

File Edit View Tools Help

Timing/Interpolation

Interpolation method : liniar

Computing time (h) = 192 µs

Range of Input = +/- 1.5V

Delay between 2 inputs (h/2) = 96 µs

Table

V	-1.5	-1	-0.5	0	0.5	1	1.5
-1.5	-3	-2.5	-2	-1.5	-1	-0.5	0
-1	-2.5	0	0.5	1	1.5	1.8	0.5
-0.5	-2	0	1	1.5	2	0.5	1
0	-1.5	0	1.5	3	2.5	1	1.5
0.5	-1	-0.5	2	2.5	2.3	1.5	2
1	-0.5	0	0.5	1	1.5	2.6	2.5
1.5	0	0.5	1	1.5	2	2.5	3

Fig. 6. GUI of the 2-Dimensional IC table

The GUI works only together with the assembly code file.

F. *The second structure*

The second structure represents the implementation of a general IC that uses 1, 2 or 3 dimension tables. The controller uses a multidimensional interpolation method with 1, 2 or 3 inputs and adjustable parameters. This implementation allows to select the type of the IC through the number of dimensions of its table. When it is necessary for the controller to exhibit a dynamic behavior, the interpolative block will be intercalated as a separable block into a structure with dynamic filters, each input and output may be filtered by a chosen filter from the library.

In Fig. 7 it is illustrated the diagram of the system and the steps until the system functions are created.

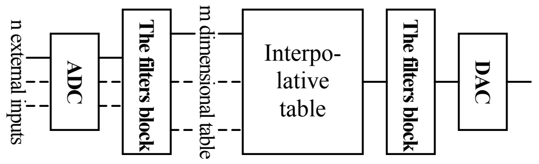


Fig. 7. General scheme of functioning

First it is specified that the interpolation table is m-dimensional (1<=m<=3). The input signal acquisition module selects and takes over the n external analog signals (1<=n<=m<=3) and after that these are converted into a digital form. The digital signals are filtered by the *input filters*. The user may choose at the moment one of the following input filters: amplifier, delay or integrator. The created filters folder gives the possibility that afterwards, other input or output filters are able to be added. If the number of the input signals is smaller than the number of the interpolation table dimensions (n<m), the remainder inputs (m-n) can be internally generated by "filtering" the selected inputs .

After an input signal is captured, converted and filtered, it will be run the algorithms described in section D. The most complex situation of searching is that in 3 dimensions. In that case, all 3 search modules are activated and all 3 indexes are provided. The program was conceived in a complex manner, thus the code describes the most difficult case (3-dimensional). If it is necessary to realize a 2-dimensional or 1-dimensional search, just a part of the base code is executed, just the corresponding modules.

In this second case the GUI have a global configure function, allowing to change the dynamical structure of the IC.

The GUI allow the following operations:

- select the number of the external inputs
- set the number of dimensions of the interpolative table
- modify the support values from the interpolative table
- choose the input filters and to put them on the input lines
- select the output filters

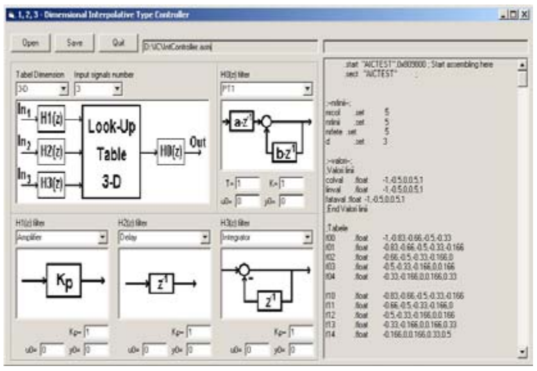


Fig. 8. GUI of the 1,2 or 3 Dimensional IC table

4. Experiments

By doing tests to verify the good system functioning, some obtained results are presented in this paper. The interpolation table is in 3-dimensions and it is ready to handle the overflow situation. This experimental table is conceived to have the output identical with the input signal. This is realized by setting the table's support values by the arithmetical average function. This function calculates the support values as the average of the corresponding support place's values.

To test the search and interpolation algorithm and also the signal acquisition, transmission and conversion all 3 external inputs are set as identical sinusoidal signals (-1V, +1V). To have the correct answer it was made a simulation in MathLab 6.5, the results were compared and found similar. The input signals were not filtered. Fig. 9 shows that the system works properly (the output is the same with the input signal).

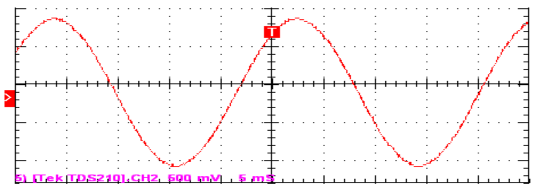


Fig. 9. The output signal in the first case.

To test more deeply the system, some modifications were made in the table (to limit down the output signal at 0V) and each input signal is filtered (2 amplifiers and 1 delay filter). Also the first input is changed to a constant signal of 0,7V.

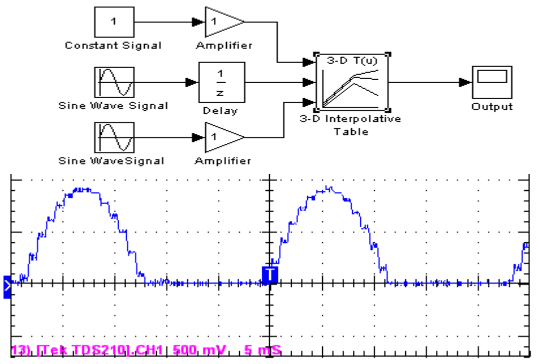


Fig. 10. The experimental structure and the output signal

The next situation presents just 1 external input but the table remains 3-dimensional, all the rest inputs internally generated by the filters. The output filter is an integrator.

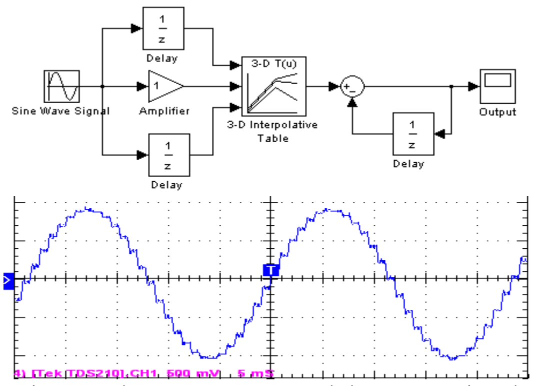


Fig. 11. The test structure and the output signal

The IC was tested for a more complicate control system for magnetic bearings of a balancing machine. The plant is a double integrator process and the controller is an interpolative 2-dimensional table implemented with DSP. The fig. 12 show the output signal obtained through simulation using MatLab and the output of the real system.

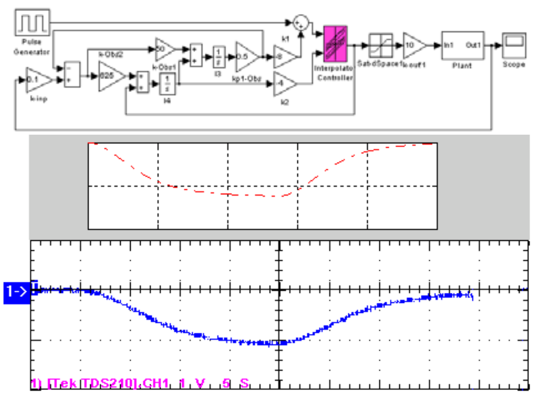


Fig. 12. The test structure and the output signal

4. CONCLUSIONS

In this project, interpolative type controllers were successfully implemented on a DSP system. From the speed's point of view, the execution time is very good (all the code instructions are executed in max. 14μs). This value of time was calculated for the most complex case when the user choose to implement an IC using a 3-dimensional table, 3 input signals and all inputs and the output filtered. If the interpolation is chosen to be in 1 or 2-dimensions the execution time is considerably diminished. Each search module lasts for 0,9μs. The interpolation module is efficiently created and last for 8μs in the most difficult case (3-dimensional interpolation). The sampling rate can be set at min. 150μs (with 3 external input signals, for each signal 50μs), because of the converter (it doesn't support a better sampling rate).

It has to be mentioned that for the TMS320C31 signal processor is not a problem (speed problem) to process a big volume of data, so, from the space point of view, only the limitation of the memory space could cause difficulties. The DSK allows the addition of external memories.

With the help of the user's interface is possible to make an essential modification in the program structure without knowing the assembly language.

5. REFERENCES

(Benchaib, 2003) Benchaib A., Rachid A., Audrezet E., "Real-Time Sliding Mode Observer and Control of an Induction Motor". *IEEE Trans. Ind. Electron,* vol. 46, pp. 128-138, Feb. 1999.

(Dragomir, 2001) Dragomir, T.L., Dale, S., Bălaş, M.M., "Some aspects regarding interpolative control", Proceedings of IWICS, 29 Nov.2001, Bucureşti.

(Kim, 2001) Kim S.H., Park T.S., Yoo J.Y., Park G.T., "Speed-Sensorless Vector Control of an Induction Motor Using Neural Network Speed Estimation". *IEEE Trans. Ind. Electron,* vol. 48, pp. 609-614, Jume 2001.

(Vukosavic, 2003) Vukosavic S.N., Levi E., "Robust DSP-Based Efficiency Optimisation of a Variable Speed Induction Motor Drive". *IEEE Trans. Ind. Electron,* vol. 50, pp. 560-570, Jume 2003.

(Youn, 2002) Youn M.J., Kim K.H., "A nonlinear Speed Control for a PM Synchronous Motor Using a Simple Disturbance Estimation Technique". *IEEE Trans. Ind. Electron,* vol. 49, pp. 525-535, Jume 2002.

(Zadeh, 2002) Zadeh, L.A. "Interpolative reasoning as o common basis for inference in fuzzy logic, neural network theory and the calculus of fuzzy If/Then rules". *Opening Talk, 2nd International conference on fuzzy logic and neural networks*, Iizuka, pp. XIII-XIV. 1992.