愛知工業大学研究報告
第３３号Ｂ　平成10年

Simulation of virtual tactile sense using DDFS circuit
DDFS 回路を用いた仮想触覚のシミュレーション

Sea-hak Kim [1]　Young-Dong Kim[2]　Seiji Hiramastu[1]　Astuo Kato[1]
金　時學，金　永炯，平松誠治，加藤厚生

[2] Chosun University, Department of control and Engineering, kwangju Korea
[1] Aichi Institute of Technology, Dept. of Electronic, Toyota, Japan

abstract
When making a high performance haptic interface, it is very important to have high safty and high position resolution. Up untill now, almost all haptic interfaces have used velocity control mode or torque control mode in order to achieve impedance control. When the servo controller of haptic interface has been used to analog circuits, there are always noise problems, in turn, affect position resolution. We tried analog viscosity/ elastic control board with filtering to solve this noise problem. But the result was also same. To solve this noise problem, it is fundamental to use digital circuits to make a high performance haptic interface which has safety and position resolution. When the servo driver is in position mode, the input signal is limitted by the number of pulses. In order to regulate velocity control mode, we have to manipulate the pulse of frequency. We fabricated a DDFS circuit to control the pulse frequency, and used this signal. We then were able to simulate simple reflect force experiment and texture sense in a digital circuit.

1. Introduction

There are 3 parts in a haptic display - a haptic interface, a rendering algorithm, and an object model. We made our haptic interface from an ideal model. We can determin the position of end-effector by using a kinematics equation, and we can solve translated degree which results from user manipulation by imploying an inverse kinematics equation and transfering this signal to a pulse. We can control velocity by changing the number of pulses throughout the DDFS circuit. When using DDFS system in position mode, the result is equal to that of torque control or velocity control. In other words, there is more protection against noise in analog board. Because it is very easy too display virtual objects and constraints planes, we did not a complex rendering algorithm.

2.1 haptic interface

Keeping the ideal paradigm in mind, we investigated the necessary components and built a high performance haptic interface resembling this ideal.

backlash - As much as possible, haptic interface must have very low backlash. Using a gear box yields unnecessary backlash. so, we used iron string like SPAIDAR and PHANToM.

torque - Haptic interface must also be capable of exerting maximum force. Because the maximum displayable force is the force that makes virtual objects, using low torque makes it difficult to display virtual wall and constraint planes.

backdrive friction - Haptic interface must have low backdrive friction. components that create backdrive friction are motor friction, transmission friction and bearing friction. We used bearing in every joint to cut down on this friction.

inertia - Haptic interface must have low inertia. When a haptic interface moves fastly or stops fastly, inertia creates unnecessary external force, and this force, in turn, gives the user an undesirable reflect force.

position resolution - Because the resolution of the encoder affects the servo rate of the haptic interface, we sought high position resolution by using 3000P/R encoder.

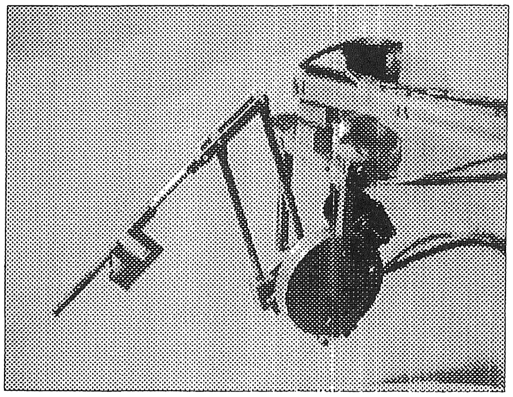


Fig 1 Extrernal preview of the haptic interface

2.2 Kinematics

We used Denavit-Hartenberg equation to compute the end-effector position of the haptic interface.

Table 1. Link Parameter of the Haptic Device

Link	αi	ai	di	θi
1	$\pi/2$	0	d1	θ_1
2	$-\pi$	L1	0	θ_2
3	$-\pi/2$	L2	0	θ_3
4	0	0	d2	0

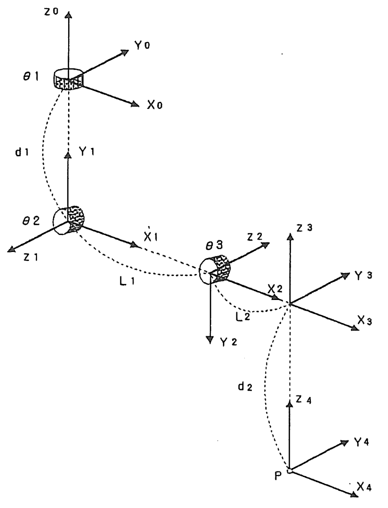


Fig 2 The kinematics representation

2.3 Inverse kinematics

If servo driver is set on position mode, the motor will not rotate, and only has load until the user inputs pulses. But every motor has a limitted amount of a displayable force, eventhough load exists, the end-effector of the haptic interface is changed slightest by the manipulation of the user. so we can determin the direction of the end-effector by investigating encoder count. Because human haptic systems have approximately 1KHz sampling resolution for tactile sense, if we can input a pulse signal with 1KHz sampling rate, the user manipulate without feeling any load. After using inverse kinematics to compute the degree, we then transfered, this degree into pulse.

2.4 DDFS(Direct Digital Frequency Synthesizer)

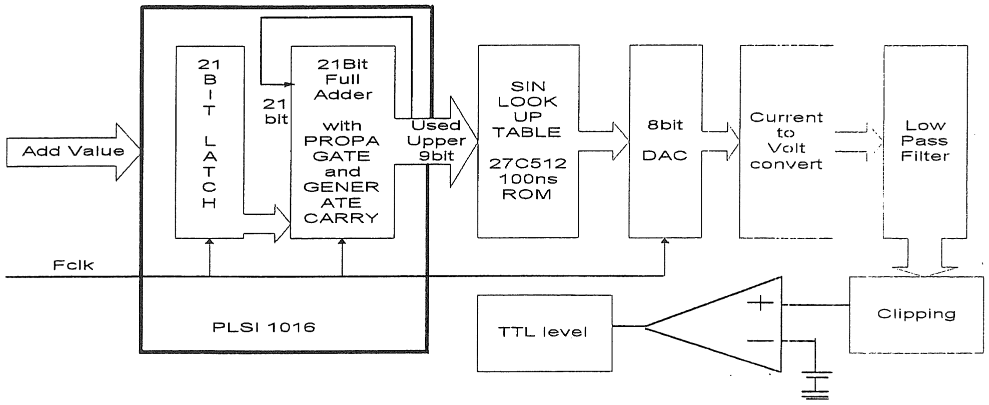


Fig 3 Block diagram of designed DDFS

We made a digital circuit which can transfer pulse number into velocity control.

In above block diagram, if the user inputs the needed pulse, then the output of the phase accumulator - phase control signal + synchronous signal - makes an address for the look-up sine table. Then reads this N bit value from the look-up sine table.

This N bit value serves as input on DAC, and DAC also outputs a sine wave which is controlled by a synchronous f_{sysclk} signal. With the f_{sysclk} set at 2MHz, we used 21bit binary width full adder in the PLSI 1016 to control the resolution of the output frequency to 1Hz. Because of high performance PLSI, ROM, and DAC, we were able to make a high quality DDFS system. Equation1 represents Nyquist law.

$$f_{\max} = f_{clkMax} \leq \frac{f_{clkmax}}{2} \qquad (1)$$

By equation 1, the maximum frequency is half of the oscillator frequency. so, the maximum frequency of this DDFS system is 1MHz. We fabricated the core of phase accumulator with PLSI which has a 21bit full adder circuit. We used 'Propagate and generate carry' generator to improve carry time. We used 512Kbit ROM with an access time of 100ns. Also we made sine value using MATLAB code pictured below transfered the output current of DAC08 into voltage.

```
i= 0:2*pi/255:pi*2;
y= 256*sin(i);
ploty(y);
grid
fid = fopen('sin.dat','wb');
count = fwrite(fid,y,'uchar')
fclose(fid)
```

Fig4 Sine value using MATLAB code

Our voltage level was about ±5V. This signal voltage becomes a sine wave after passing through the Low Pass Filter. We can then convert this sine wave into a 'TTL level' pulse by clipping any signal above 0V with zener diode and comparator. We used this signal to control velocity.

2.5 Rendering algorithm

Second component to do haptic display is rendering algorithm. Almost all haptic rendering algorithm have complex equations for end-effector of haptic interface not to penertrate into virtual objects or to compute direction of normal vector. So, they have limit of polygons number to display virtual object in realtime. There are various rendering algorithms-the vector field method, the god-object algorithm and so on. In case of god-object algorithm, running on a 66MHz pentium PC, it is capable of rendering objects with up to about 1000 surfaces at a servo rate of about 1KHz. 「4」 But in position control mode, we need only position of collision detection to display virtual wall. So, rendering algorithm will be simple. We used god-object algorithm to compute position of collision detection on a virtual object. X,Y,Z represent position of collision detection on a virtual object and Px,Py,Pz represent position of haptic interface. The new position of collision detection on a virtual wall is found by minimizing L in equation 2 below by setting all six partial derivatives of L to 0.

$$L=\frac{1}{2}(Px-X)^2 + \frac{1}{2}(Py-Y)^2+\frac{1}{2}(Pz-Z)^2 + A_1X+B_1Y+C_1Z-D_1 + A_2X+B_2Y+C_2Z-D_2 + A_3X+B_3Y+C_3Z-D_3 \quad (2)$$

3 Experiment

1) Making free state

To achieve a free state in the end-effector of our haptic interface, is a very important experiment for this position control. Using a DDFS circuit, we experimented to create free state in the end-effector of our haptic interface. We connected one pentium 166MHz computer with the haptic interface and the DDFS circuit, and simulated a virtual object. But we couldn't create a perfect free state, because of a time delay.

2) Making virtual wall

The results from our experiments to create virtual wall are entered below in Fig 5. We took this data only when the user could feel a free state in the end-effector of the haptic interface.

The region from 0 to 30 is a free state. Past 30, the haptic interface created a stiff virtual wall.

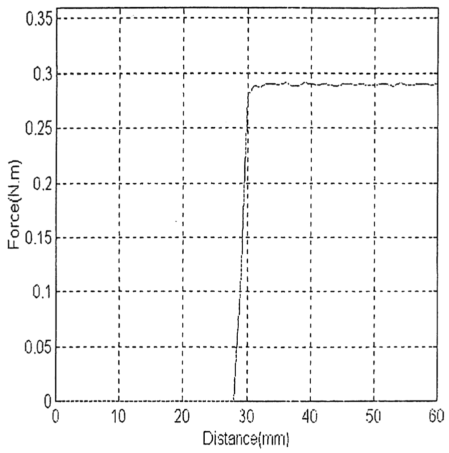


Fig 5 The grapgh of virtual wall

3) Virtual surface display

Using this model - the extension of the perceptual observation that sideways spring forces feel like gravity acting on the hand. Magaret Minsky made sandpaper system. In this paper, we used local geometry to display virtual surfaces including rough and bumpy. We recorded the feelings of 11 people who are volunteers in our laboratory - 10 men, 1 woman, 9 right handed, 2 left handed, with an average age 21.2. Because we couldn't create a perfect free state in the end-effector, all the sensations that experienced were determined by a pre-programmed. We simulated 9 signals, and our results are listed below in table 2.

Table.2 9 input signals in our experiments

	HWP	VWP	High	++	+	-
1	8	4	4	7	4	0
2	8	8	4	6	4	1
3	8	20	4	3	8	0
4	20	4	4	5	5	1
5	20	8	4	4	6	1
6	20	20	4	6	5	0
7	40	4	4	3	8	0
8	40	8	4	5	5	1
9	40	20	4	11	0	0

HWP means Hill Width Pulse
VWP means Valley Width Pulse
++ means confidence between 70 - 100%
+ means confidence between 40 - 70%
- means confidence between 0 - 40%

4. Conclusion and Future work

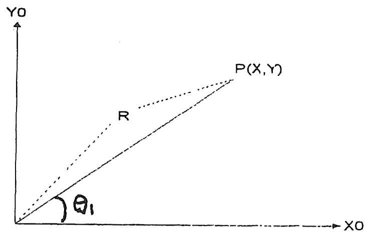
In this paper, we controlled velocity using a DDFS system in position control mode. We created a DDFS system which has a high resolution frequency generator, and programmed. We simulated simple reflect force and texture sense using a geometric method. We present that it is possible to have virtual tactile sense using a DDFS system and a haptic system in position control mode. In future work, we will use a DSP(Digital Signal Processor) to reduce the time required in progressing both the necessary complex equation and the complex algorithms which in previous experiments had made the system servo rate very slow.

References

(1) Massie, T. H. : *Design of a Three DegreeDi of Freedom Force-Reflecting Haptic Interface.* M.S. Thesis, Dept. of Electrical Engineering, MIT, 1993.
(2) Massie, T. H. and Salisbury, J. K. : The PHANToM Haptic Interface : A Device for Probing Virtual Objects. *Proc. of ASME Winter Annual Meeting*, Chicago, 1994.
(3) Massie, T. H. : Virtual Touch Through Point Interaction. *ICAT'96.* 19-38, 1996.
(4) Craig B. Zilles : Haptic Rendering with the Toolhandle Haptic Interface. M.S. Thesis, Dept. of Mechanical Engineering, MIT, 1995.
(5) Minsky, M. : *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display.* Ph.D. dissertation, MIT, 1995.
(6) Iwata, H.: Pen Based Haptic Virtual Environment. *VRAIS'93.* 287-292, 1993.

Appendix

Results of inverse kinematics of the haptic interface



$R=\sqrt{Px^2+Py^2}$, $\sin\theta_1=\frac{Py}{R}$, $\cos\theta_1=\frac{Px}{R}$

$\therefore \theta_1=atan2(Px,Py)$

$OA=\sqrt{Px^2+Py^2}$

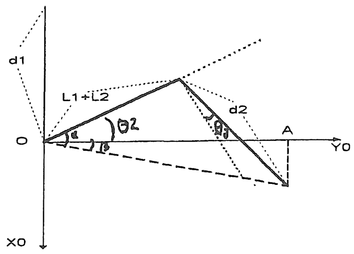
$AP=P_z-d_1$

$OP=\sqrt{Px^2+Py^2+(P_z-d_1)^2}$

In case θ_2 is rotating (-) direction and θ_3 is rotating (-) direction

$$\cos\varphi=\frac{(L_1+L_2)^2+d_2{}^2-OP}{2(L_1+L_2)d_2}=A$$

$\sin\varphi=\sqrt{1-\cos\varphi^2}$



$$\tan\varphi = \frac{\sqrt{1-\cos\varphi^2}}{A}$$

$$\therefore \varphi = atan2(A, \sqrt{1-\cos\varphi^2})$$

$$\therefore \theta_3 = \varphi - 90$$

$$\cos\alpha = \frac{OA}{OP}, \ \sin\alpha = \frac{AP}{OP}$$

$$\tan\alpha = \frac{AP}{OA}$$

$$\therefore \alpha = atan2(OA, AP)$$

$$\cos\beta = \frac{(L_1+L_2)^2 + OP^2 - d_2{}^2}{2(L_1+L_2)OP} = B$$

$$\sin\beta = \sqrt{1-\cos^2\beta}$$

$$\therefore \tan\beta = \frac{\sqrt{1-\cos^2\beta}}{A}$$

$$\therefore \beta = atan2(A, \sqrt{1-\cos^2\beta})$$

$$\therefore \theta_2 = \alpha - \beta = atan(OA, AP) - atan(A, \sqrt{1-\cos^2\beta})$$

In case θ_2 is rotating (−) direction and θ_3 is rotating (+) direction

$$\therefore \theta_3 = 90 - \varphi$$

$$= 90 - atan2(A, \sqrt{1-\cos\varphi^2})$$

$$\therefore \theta_2 = \alpha - \beta = atan2(OA, AP) - atan2(A, \sqrt{1-\cos^2\beta})$$

In case θ_2 is rotating (−) direction and θ_3 is rotating (−) direction

$$\therefore \theta_3 = \varphi - 90 = atan2(A, \sqrt{1-\cos\varphi^2}) - 90$$

$$\therefore \theta_2 = \alpha - \beta = atan2(A, \sqrt{1-\cos^2\beta}) - atan2(OA, AP)$$

In case θ_2 is rotating (−) direction and θ_3 is rotating (+) direction

$$\therefore \theta_3 = 90 - \varphi$$

$$= 90 - atan2(A, \sqrt{1-\cos\varphi^2})$$

$$\therefore \theta_2 = \alpha - \beta = atan2(A, \sqrt{1-\cos^2\beta}) - atan2(OA, AP)$$

The results of kinematics equations

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} C_1C_2(L_1+L_2C_3+d_2S_3)+C_1S_2(L_2S_3-d_2C_3) \\ S_1C_2(L_1+L_2C_3+d_2S_3)+S_1S_2(L_2S_3-d_2C_3) \\ S_2(L_1+L_2C_3+d_2S_3)-C_2(L_2S_3-d_2C_3)+d_1 \end{bmatrix}$$

（受理　平成10年 3 月20日）