เอกสารอ้างอิง

Kuo, B. C., Automatic Control System, Prentice-Hall of India Private Limited, New Delhi, 5th ed., 1987.

Ogata, K., Modern Control Engineering, Prentice-Hall of India Private Limited, New Delhi, 1984.

Palm, W. J., Modeling, Analysis, and Control of Dynamic Systems, John Wiley and Sons, Inc., New York, 1983.

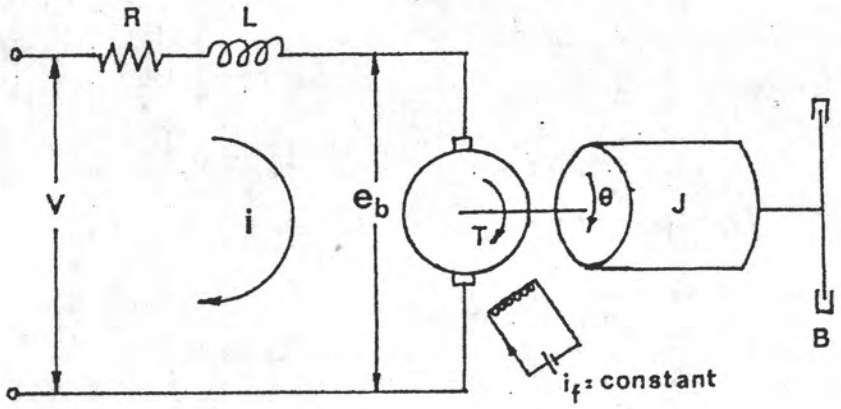
Tomizuka, M., D. Dornfeld, and M. Purcell, "Application of Microcomputers to Automatic Weld Quality Control," ASME Journal of Dynamic System, Measurement, and Control, 102, 62-68, 1980.

Tomizuka, M., D. Dornfeld, X. Q. Bian, and H. G. Cai, "Experimental Evaluation of the Preview Servo Scheme for a Two-Axis Position System," ASME Journal of Dynamic Systems, Measurement, and Control, 106, 1-5, 1984.

ภาคผนวก

ภาคผนวก ก.
แบบจำลองทางคณิตศาสตร์ของมอเตอร์กระแสตรง

มอเตอร์กระแสตรงเป็นอุปกรณ์ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล ซึ่งนิยมใช้ในระบบควบคุมทั่วๆไป สำหรับโครงการวิทยานิพนธ์นี้ใช้มอเตอร์กระแสตรงแบบแม่เหล็กถาวร



รูปที่ 1 วงจรของมอเตอร์กระแสตรงแบบฟิลด์คงที่

กำหนดให้

R = ความต้านทานของอาร์มาเจอร์
L = อินดัคแต้นซ์ของอาร์มาเจอร์
i = กระแสอาร์มาเจอร์
i_f = กระแสของฟิลด์
V = โวลท์เต็จที่ป้อนให้กับมอเตอร์
e_b = โวลท์เต็จย้อนกลับ(back emf)
θ = มุมที่มอเตอร์หมุน
T = แรงบิดที่ได้จากมอเตอร์
J = โมเมนต์ของแรงเฉื่อยของมอเตอร์และโหลด
B = สัมประสิทธิ์ของวิสคอสแด้มปิ้ง ของมอเตอร์และโหลด

เนื่องจากมอเตอร์ที่ใช้เป็นแบบแม่เหล็กถาวร เส้นแรงของฟิลด์มีค่าคงที่ ดังนั้นอัตราส่วนระหว่างกระแสอาร์มาเจอร์และและแรงบิดจะมีค่าคงที่ ตามสมการที่ 1

$$T = K_m \cdot i \qquad (1)$$

เมื่อ K_m = ค่าคงที่แรงบิดของมอเตอร์ (motor-torque constant) และในขณะที่มอเตอร์หมุนจะเกิดโวลท์เต็จย้อนกลับ ซึ่งมีค่าเท่ากับ

$$e_b = K_b \cdot d\theta/dt \qquad (2)$$

เมื่อ K_b = ค่าคงที่ของโวลท์เต็จย้อนกลับของมอเตอร์ จากรูปที่ 1 เราจะได้ว่า

$$L \cdot di/dt + R \cdot i + e_b = V \qquad (3)$$

$$J \cdot \theta + B \cdot \theta = T = K_m \cdot i \qquad (4)$$

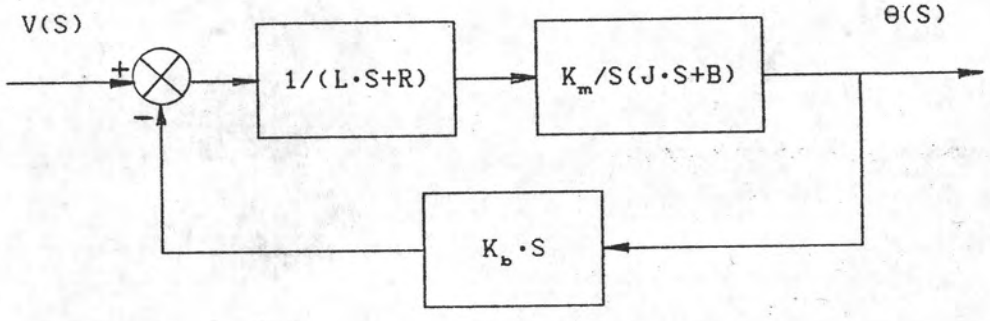
ถ้ากำหนดให้เงื่อนไขเริ่มต้นเป็นศูนย์ จากสมการที่ (2), (3) และ (4) ใส่ลาปลาสทรานสฟอร์มทั้งสองข้าง จะได้สมการใหม่เป็น

$$K_b \cdot S \cdot \theta(S) = E_b(S) \qquad (5)$$

$$(L \cdot S + R) I(S) + E_b(S) = V(S) \qquad (6)$$

$$(J \cdot S^2 + B \cdot S)\theta(S) = T(S) = K_m \cdot I(S) \qquad (7)$$

จากสมการที่ (5),(6) และ (7) เราสามารถเขียนเป็นบล็อคไดอะแกรม ดังแสดงในรูปที่ 2 และทำการยุบบล็อค จะได้ทรานสเฟอร์ฟังชันตามสมการที่ 8



รูปที่ 2 บล็อคไดอะแกรมของวงจร

$$\theta(S)/V(S) = K_m/S[L \cdot J \cdot S^2 + (L \cdot B + R \cdot J)S + R \cdot B + K_m \cdot K_b] \quad (8)$$

ถ้าเราถือว่า L มีค่าน้อยมากและตัดทิ้งได้ จะได้ว่า

$$\theta(S)/V(S) = K/S(T_m \cdot S+1) \quad (9)$$

กำหนดให้

$$K = K_m/(R \cdot B + Km \cdot K_b) \quad (10)$$

$$= \text{motor gain constant}$$

$$T_m = R \cdot J/(R \cdot B + K_m \cdot K_b) \quad (11)$$

$$= \text{motor time constant}$$

ภาคผนวก ข.

รายละเอียดของมอเตอร์ที่ใช้ในการทดลอง

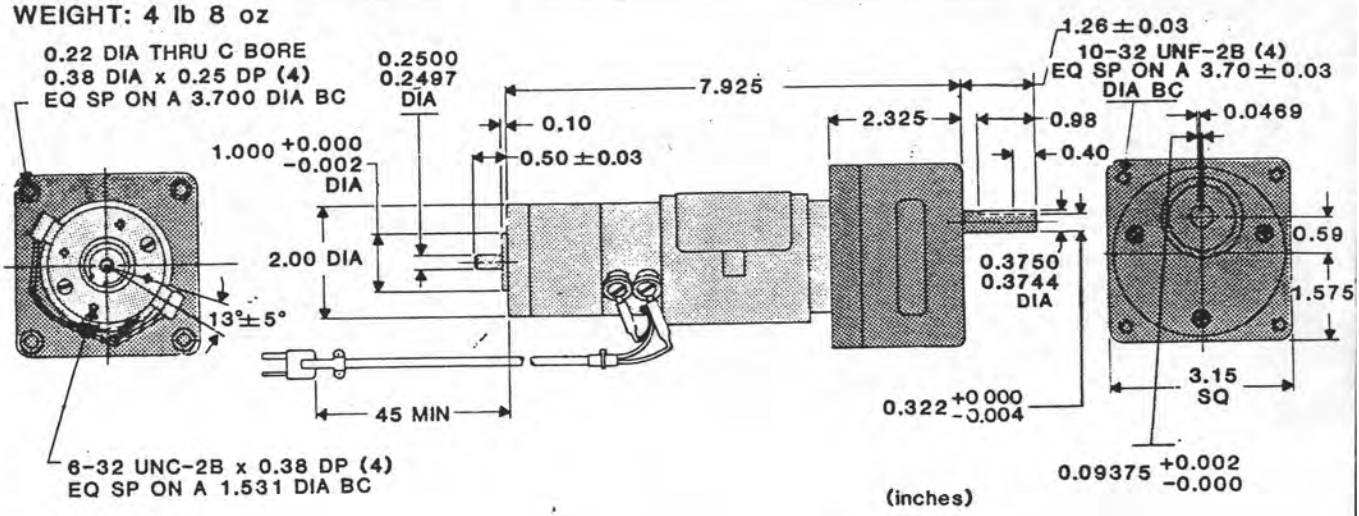
มอเตอร์ที่ใช้ในโครงการวิทยานิพนธ์นี้เป็นมอเตอร์กระแสตรงของบริษัท Electro Craft Model 586-022-113 ด้านท้ายของมอเตอร์มี Tachometer ด้านเพลามอเตอร์มีชุดเฟืองทดที่มีอัตราทด 100:1 สำหรับรายละเอียดต่างๆ ของมอเตอร์แสดงดังตารางต่อไปนี้

SPECIFICATIONS MODELS

			E586-MG
Rated Voltage	V_r	V	36
NO-LOAD SPEED at V_r	N_o	rpm	6200
MAX. RATED I at STALL	I_r	A	4.6
STALL TORQUE at I_r	T_o	oz-in	29
MAX. PULSE CURRENT	I_p	A	24.0
TORQUE CONSTANT	K_t	oz-in/A	7.8
VOLTAGE CONSTANT	K_e	volts/krpm	5.8
TERMINAL RESISTANCE	R_t	Ωat 25°C	1.1
Armature Mom. Inertia	J_m	oz-in-s^2	5.5x10^{-2}
Rotational Loss Constant	K_d	oz-in/krpm	0.10
Static Friction Torque	T_f	oz-in	3.0
Thermal Resistance Arm./Amb.	R_{th}	°C/W	5.0
ARMATURE INDUCTANCE	L_a	mH	2.3
ELECTRICAL Time Constant	τ_e	ms	2.1
MECHANICAL Time Constant	τ_m	ms	14.0
TACH. Voltage Gradient	K_g	V/krpm	14.2
TACH. Terminal Resistance	R_g	Ωat 25°C	720
TACH. Armature Inductance	L_g	mH	138
TACH. Load Resistance (optimum)	R_l	Ω	5000
Ripple Amplitude		%pk-pk	5.0
LINEARITY		%	0.2
Temperature Coefficient		%/°C	-0.05

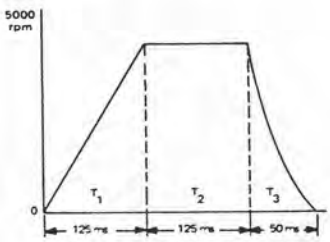
E586-MGHP 586-022-XXX

WEIGHT: 4 lb 8 oz



E586-MG Matching Motor Generator

PART NUMBER 0586-00-028



Motor Type	0586-00-028
Control Type	E586-M
†Load Inertia	0.011 oz-in-s²
	5000 rpm

†Load inertia is defined as the total moment of inertia
J_T J_M+J_L J_M(motor-generator) is .0055 oz-in-s²
J_L (load) is .0055 oz-in-s²

Features

Hardened stainless steel shaft.

Die case end caps.

Plated tubular steel housing.

Armature windings are epoxy coated and dynamically balanced. The motor and generator commutators are machined to a fine micro-finish (20-30) to insure proper brush-commutator interface.

High energy product (grade 8) magnets withstand higher demagnetization currents.

Ball bearings are shielded to maximize reliability and long life.

External brush holders facilitate motor and generator brush replacement.

Metal graphite brushes withstand high pulse current and maximize brush life.

Separate motor and generator armature.

Temperature compensated generator.

Bi-directional.

Machined mounting surface on all flange mounting models.

Rated at 1/10 hp with matching E586 servo motor control.

E586-MGHP Matching Gearmotors

Ratio	PART NUMBER	Max. Speed	Max. Torqu
6:1	586-022-102	833 rpm	6 lb-in
12.5:1	586-022-103	400 "	12 "
30:1	586-022-107	167 "	26 "
50:1	586-022-109	100 "	44 "
100:1	586-022-113	50 "	79 "
150:1	586-022-115	33 "	90 "
180:1	586-022-116	28 "	90 "

Flange mounted — all steel gears — ball bearings — precisio die cast housing.

Lubricant Grease

Maximum thrust load 15 lb.

Maximum overhung load 25 lb ½" from face.

ภาคผนวก ค.
การตัดแผ่นเหล็กด้วยเปลวไฟ

ในการตัดแผ่นเหล็กด้วยเปลวไฟโดยใช้ก๊าซอะซิทิลีนและก๊าซออกซิเจน ความร้อนที่ตัดเหล็กให้ขาดได้มาจาก การที่เหล็กร้อนแดงทำปฏิกริยากับออกซิเจน แล้วให้เหล็กออกไซด์กับพลังงานความร้อน 500 MJ ซึ่งพลังงานความร้อนนี้จะทำให้เหล็กหลอมละลายและถูกเป่าให้หลุดออกไปด้วยแรงดันของก๊าซออกซิเจน

สำหรับความสัมพันธ์ระหว่างความเร็วการตัด ขนาดของแผ่นเหล็กและปริมาณของก๊าซอะซิทิลีนและก๊าซออกซิเจนแสดงไว้ในตารางดังต่อไปนี้

Table 2. Typical Data for Manual Oxyacetylene Cutting of Low-Carbon Steel Plate

Plate thickness, in.	Diameter of cutting orifice, in.	Oxygen pressure, psi	Cutting speed, ipm(a)	Gas consumption, cu ft(b)(c)			
				Per hr		Per linear ft	
				Oxygen	Acetylene	Oxygen	Acetylene
1/8	0.0380-0.0400	15-23	20-30	45-55	7-9	0.37-0.45	0.06-0.07
1/4	0.0380-0.0595	11-20	16-26	50-93	9-11	0.63-0.72	0.08-0.11
3/8	0.0380-0.0595	17-25	15-24	60-115	10-12	0.80-0.96	0.10-0.13
1/2	0.0465-0.0595	20-30	12-22	66-125	10-13	1.10-1.14	0.12-0.17
3/4	0.0465-0.0595	24-35	12-20	117-143	12-15	1.43-1.95	0.15-0.20
1	0.0465-0.0595	28-40	9-18	130-160	13-16	1.78-2.89	0.18-0.29
1 1/2	0.0595-0.0810	35-48	6-14	143-178	15-18	1.96-3.18	0.21-0.33
2	0.0670-0.0810	22-50	6-13	185-231	16-20	3.55-6.16	0.31-0.53
3	0.0670-0.0810	33-55	4-10	240-290	19-23	5.80-12.00	0.46-0.95
4	0.0810-0.0860	42-60	4-8	293-388	21-26	9.70-14.64	0.65-1.05
5	0.0810-0.0860	53-70	3.5-6.4	347-437	24-29	13.66-19.83	0.91-1.37
6	0.0980-0.0995	45-80	3.0-5.4	400-567	27-32	21.00-26.70	1.19-1.80
8	0.0995	60-77	2.6-4.2	505-615	31.5-38.5	29.30-38.84	1.83-2.42
10	0.0995	75-96	1.9-3.2	610-750	36.9-45.1	46.90-64.20	2.57-3.84
12(c)	0.1200	69-86	1.4-2.6	720-880	42.3-51.7	67.70-103.00	3.98-6.05

(a) Lowest speeds and highest gas consumptions are for inexperienced operators, short cuts, dirty or poor material. Highest speeds and lowest gas consumptions are for experienced operators, long cuts, and clean and good material.

(b) Because the pressure of acetylene for the preheating flames is more a function of torch design than of the thickness of the part being cut, pressure data have been omitted from this table. For acetylene pressure data, see charts of manufacturers of apparatus.

(c) Beyond 12-in. thickness, the critical data of manual cutting practices are greatly affected by the condition of the metal and the skill of the operator, resulting in wide ranges of data. In view of this, the table has been terminated at the 12-in. thickness. In Table 3, on machine cutting, thickness range is extended to 36 in.

Table 3. Typical Data for Machine Oxyacetylene Cutting of Low-Carbon Steel Plate(a)

Plate thickness, in.	Diameter of cutting orifice, in.	Oxygen pressure, psi	Cutting speed, ipm(b)	Gas consumption, cu ft(b)(c) Per hr Oxygen	Per hr Acetylene	Per linear ft Oxygen	Per linear ft Acetylene
⅛	0.0250-0.0400	15-23	22-32	40-55	7-9	0.34-0.36	0.05-0.06
¼	0.0310-0.0595	11-35	20-28	45-93	8-11	0.34-0.66	0.07-0.08
⅜	0.0310-0.0595	17-40	19-26	82-115	9-12	0.86-0.89	0.08-0.09
½	0.0310-0.0595	20-55	17-24	105-125	10-13	1.04-1.24	0.11-0.12
¾	0.0380-0.0595	24-50	15-22	117-159	12-15	1.45-1.56	0.14-0.16
1	0.0465-0.0595	28-55	14-19	130-174	13-16	1.83-1.86	0.17-0.19
1½	0.0670-0.0810	25-55	12-15	185-240	14-18	3.20	0.23-0.24
2	0.0670-0.0810	22-60	10-14	185-260	16-20	3.70-3.72	0.29-0.32
3	0.0810-0.0860	33-50	8-11	240-332	18-23	6.00-6.04	0.42-0.45
4	0.0810-0.0860	42-60	6.5-9	293-384	21-26	8.53-9.02	0.58-0.65
5	0.0810-0.0860	53-65	5.5-7.5	347-411	23-29	10.97-12.62	0.77-0.84
6	0.0980-0.0995	45-65	4.5-6.5	400-490	26-32	15.10-17.78	0.98-1.16
8	0.0980-0.0995	60-90	3.7-4.9	505-625	31-39	25.52-27.30	1.59-1.68
10	0.0995-0.1100	75-90	2.9-4.0	610-750	37-45	37.50-42.10	2.25-2.55
12	0.1100-0.1200	69-105	2.4-3.5	720-880	42-52	49.70-60.00	2.97-3.50
12	0.1935	30	(e)	1274	85	(d)	(d)
14	0.221	25	(e)	1458	98	(d)	(d)
16	0.221	30	(e)	1683	98	(d)	(d)
18	0.250	25	(e)	1838	134	(d)	(d)
20	0.250	30	(e)	2098	134	(d)	(d)
22	0.250	35	(e)	2358	134	(d)	(d)
24	0.290	25	(e)	2467	174	(d)	(d)
26	0.290	30	(e)	2772	174	(d)	(d)
28	0.290	35	(e)	3077	174	(d)	(d)
30	0.332	25	(e)	3125	204	(d)	(d)
32	0.332	30	(e)	3425	204	(d)	(d)
34	0.332	35	(e)	3775	204	(d)	(d)
36	0.332	40	(e)	4125	204	(d)	(d)
Cutting With One-Piece Divergent Nozzles (High-Speed Tips)(f)							
¼	...	120	26	70	14	0.54	0.109
½	...	100	22	100	18	0.91	0.163
¾	...	100	20	120	18	1.00	0.18
1	...	110	18	130	18	1.44	0.20
1½	...	100	16	170	25	2.11	0.31
2	...	100	13	230	25	3.52	0.39
3	...	85	10	270	32	5.4	0.64
4	...	85	9	230	35	7.3	0.78
5	...	110	8	395	35	9.7	0.87
6	...	125	7	435	35	12.4	1.00
7	...	80	6	495	36	16.5	1.20
8	...	100	5½	580	36	21.2	1.32

(a) Column values do not necessarily vary in exact proportion to plate thickness, because straight-line relations do not exist among pressure, speed, and orifice sizes.

(b) Lowest speeds and highest gas consumptions are for inexperienced operators, short cuts, dirty or poor material. Highest speeds and lowest gas consumptions are for experienced operators, long cuts, clean and good material.

(c) Because the pressure of acetylene for the preheating flames is more a function of torch design than of the thickness of the part being cut, pressure data have been omitted from this table. For acetylene pressure data, see charts of manufacturers of apparatus.

(d) The data for 12 to 36 in. are for torches with large tips, using large volumes of oxygen at low pressures. Because the speed depends on the condition of the material being cut and the skill of the operator, no figures are given for gas consumption per linear foot.

(e) Cutting speeds vary from 4 ipm for light sections to 2 ipm for heavy sections.

(f) These data are for divergent orifices that have controlled expansion tips, permitting the use of relatively high oxygen pressures so as to expand the cutting oxygen through the orifices. Thus the gas stream has high velocity and is relatively narrow. Higher speeds and lower specific consumption of oxygen result.

ภาคผนวก ง.
โปรแกรมที่ใช้ในการควบคุม

โปรแกรมที่เขียนขึ้นเพื่อใช้ในการควบคุมเขียนด้วยภาษา C และใช้ complier Optimizing C86 ver 2.20J ของบริษัท COMPUTER INNOVATIONS, INC. ซึ่งมีรายละเอียดของโปรแกรมดังต่อไปนี้

```
                /*   mctrl.c  16:10:55  11/15/1987   */

#include "stdio.h"

 mctrl(kgain,totalp,numpoint,pvp,vlimit,f2)
  float kgain[][3],vlimit;
  int   totalp,numpoint,pvp,f2;
   {
    extern double fabs();
    extern float data[][10],sqrt();
    int chnox,chnoy,i,j,idelay,index;
    long xm,ym;
    float tempp,errx,erry,vdax,vday,voltx,volty,zero;
    float xc,yc,errxc,erryc,xs,ys,vx,vy,vcalx,vcaly;
    int start,delay2,endd;
    extern key_getc();
    char ch;
     chnox = 1;          /* Read vel x chno. 1 */
     chnoy = 2;          /* Read vel y chno. 2 */
     index = 0;          /* start index        */
     zero  = 0;
     errx  = 0;
     erry  = 0;
     vx    = 0;
```

```
vy     = 0;
vcalx = 0;
vcaly = 0;
xs     = data[index][1]*1000;
ys     = data[index][2]*1000;
crt_cls();
block1(0,0,23,79);
crt_srcp(12,10,0);
printf("Calibrate Velocity");
for(i=1;i<=1000;i++)
 {
  readv(&voltx,chnox);
  readv(&volty,chnoy);
  vcalx += voltx;
  vcaly += volty;
 }
vcalx = vcalx/1000.0;
vcaly = vcaly/1000.0;
crt_srcp(12,10,0);
printf("Go to Start Point ");
go_tar(xs,ys);
crt_srcp(12,10,0);
printf("Press Any Key to Begin  Control or Esc to Cancel");
delay2 = 2.5*1230L * 10;
start = -1;
do
{
 sound(2,10);
 for(i=0;i< delay2;i++);
 start = key_scan();
}while (start == -1);
```

```
if (start != 283)
{
 crt_srcp(12,10,0);
 printf("                    Begin Control                    ");
 crt_srcp(12,40,0);
/* begin controller */
readp(&xm,&ym);
tempp = xm;
data[index][3] = tempp/100000;
tempp = ym;
data[index][4] = tempp/100000;
readv(&voltx,chnox);
readv(&volty,chnoy);
data[index][5] = (vcalx - voltx)*0.0525;
data[index][6] = (vcaly - volty)*0.0391;
data[index][8] = data[index][9] = 0 ; /* torque start    */
for(index=1;index<=numpoint;index++)
 {
   gain(kgain,index,pvp,errx,erry,vx,vy,sampling);
   vdax = (data[index][8] - 0.03115)/(3.982691);
   vday = (data[index][9] - 0.04298)/(3.976296);
   data[index][8] = vdax;
   data[index][9] = vday;
   if (vdax >= vlimit || vdax <= -1*vlimit ||
       vday >= vlimit || vday <= -1*vlimit )
      {
       out(zero,zero);
       crt_srcp(12,10,0);
       printf("Volt Out Excess limit Press Any Key ");
       index = numpoint;
       start = key_getc();
```

```
        start = 0;
      }
   else
      out(vdax,vday);
   for(idelay = 1;idelay <=endd;idelay++);   /* delay loop */
   readp(&xm,&ym);
   readv(&voltx,chnox);
   readv(&volty,chnoy);
   tempp = xm;
   data[index][3] = tempp/100000;
   tempp = ym;
   data[index][4] = tempp/100000;
   vx = (data[index][3] - data[index-1][3])/sampling;
   vy = (data[index][4] - data[index-1][4])/sampling;
                          /* update 20-10-87  */
   data[index][5] = (vcalx - voltx)*0.0525;
   data[index][6] = (vcaly - volty)*0.0391;
   errx += (data[index][1]-data[index][3])*sampling;
   erry += (data[index][2]-data[index][4])*sampling;
}
out(zero,zero);
for(i=0;i<=totalp;i++)
 {
 if (f2 == 1)       /*  keep Error */
  {
  data[i][8] = (data[i][1]-data[i][3])*1000;
  data[i][9] = (data[i][2]-data[i][4])*1000;
  }
 data[i][7] =  sqrt(data[i][5]*data[i][5] +
               data[i][6]*data[i][6]);
 }
```

```
                    /*  gain       14:35:09  11/14/1987 */

gain(kgain,ref,pvp,errx,erry,vx,vy,sampling)
float kgain[][3],errx,erry,vx,vy,sampling;
int   pvp,ref;

 {
   extern float data[][10];
   float  tpx,tix,tdx,tprx,vdx;
   float  tpy,tiy,tdy,tpry,vdy;
   int    act;
      act  = ref -1;
      vdx  = (data[ref][1]-data[act][1])/sampling;
      vdy  = (data[ref][2]-data[act][2])/sampling;

      tpx  = kgain[1][1]*(data[ref][1]-data[act][3]);
      tix  = kgain[2][1]*errx;
      tdx  = kgain[3][1]*(vdx-vx);
      tprx = kgain[4][1]*(data[ref+pvp][1]-data[act][3]);
      data[ref][8] = tpx + tix + tdx + tprx ;

      tpy  = kgain[1][2]*(data[ref][2]-data[act][4]);
      tiy  = kgain[2][2]*erry;
      tdy  = kgain[3][2]*(vdy-vy);
      tpry = kgain[4][2]*(data[ref+pvp][2]-data[act][4]);
      data[ref][9] = tpy + tiy + tdy + tpry ;


 }

                              /* End gain */
```

```
 printf("Begin Simulation :");
 for(index=1;index<=numpoint;index++)
   {
    gain(kgain,index,pvp,errx,erry,vx,vy,sampling);

    model(index,sampling);

    vx = (data[index][3] - data[index-1][3])/sampling;
    vy = (data[index][4] - data[index-1][4])/sampling;
    errx += (data[index][1]-data[index][3])*sampling;
    erry += (data[index][2]-data[index][4])*sampling;
    crt_srcp(12,28,0);
    printf("%d",index);

   }
   for(i=0;i<=totalp;i++)
    {
     if (f2 == 1)                    /*  keep Error */
      {
       data[i][8] = (data[i][1] - data[i][3])*1000;
       data[i][9] = (data[i][2] - data[i][4])*1000;
      }
     data[i][7] =  sqrt(data[i][5]*data[i][5] + data[i][6]*data[i][6]);
    }
}

                /*        end sim.c        */
```

```
                /*      sim.c   17:04:25   11/14/1987  */


#include "stdio.h"

 simu(kgain,totalp,numpoint,pvp,f2,sampling)
  float kgain[][3],sampling;
  int   totalp,numpoint,pvp,f2;
   {

    extern float data[][10],sqrt();
    int i,index;
    float errx,erry,vx,vy;

     crt_cls();
     block1(0,0,23,79);
     index = 0;
     errx  = 0;
     erry  = 0;
     vx    = 0;
     vy    = 0;

     /* begin controller */

     data[index][3] = data[index][1];
     data[index][4] = data[index][2];

     data[index][5] = data[index][6] = 0 ; /* start velocity */
     data[index][8] = data[index][9] = 0 ;  /* torque */

     crt_srcp(12,10,0);
```

```
                /*      model.c     14:54:29  11/14/1987 */

#include "stdio.h"
            /*  define Parameter modelx  */

#define massx 11.423
#define jx    2.0*6.18960849e-4
#define bx    6.118242512e-5

           /* define parameter modely */

#define massy 2.174
#define jy    2.0*6.18960849e-4
#define by    6.118242512e-5
#define g     9.81
#define mu    0.02*8.0
#define rpull 0.025
#define R     1.1                   /* Motor Amature resistance */
#define L     2.3e-3                /* Motor Inductance         */
#define Km    7.8*7.0612e-3         /* Motor Torque Constant    */
#define n     100.0                 /* Motor Gear Ratio         */
#define jm    5.5e-3*3.499e-3       /* Motor Mom. Intia         */
#define stm   6*7.0612e-3           /* Static Friction Torque   */

model(point,sampling)
int point;
float sampling;

{
extern double exp();
extern float data[][10];
```

```
float cx1,cx2,cx3,cx33,cx4,sx,wx,toqx,toqy;
float cy1,cy2,cy3,cy33,cy4,sy,wy;

  toqx = (data[point][8]/19.9719086)*35.0;
  cx33 = mu*rpull*massx*g;
  cx1  = (jx+rpull*rpull*massx +jm*n*n);
  cx2  = bx*n;
  cx33 = cx33 + stm;
  if (toqx <= cx33) cx33 = 0.0;
  if (toqx< 0) cx33 = -cx33;
  cx3  = (toqx - cx33);
  cx4  = exp(-1.0*sampling*cx2/cx1);
  sx   = data[point-1][3]/rpull; /* seta motion */
  wx   = data[point-1][5]/rpull ;    /* omega       */
  data[point][3]  = rpull*((cx1*cx3-cx1*cx2*wx)*cx4+cx2*cx3*sampling
                    - cx1*cx3+cx1*cx2*wx+cx2*cx2*sx)/(cx2*cx2);
  data[point][5] = rpull*(wx+(cx1*cx3-cx1*cx2*wx)*(1-cx4)/(cx1*cx2));

            /*  End modelx */

             /*    modely   */
  toqy = (data[point][9]/19.9719086)*35.0;
  cy33 = mu*rpull*massy*g;
  cy1  = (jy+rpull*rpull*massy + jm*n*n);
  cy2  = by*n;
  cy33 = cy33 + stm;
  if (toqy < cy33) cy33 = 0.0;
  if (toqy < 0) cy33 = -cy33;
  cy3  = (toqy - cy33);
  cy4  = exp(-1.0*sampling*cy2/cy1);
  sy   = data[point-1][4]/rpull; /* seta motion */
```

```
  wy   = data[point-1][6]/rpull ;  /*  omega       */
  data[point][4]  = rpull*((cy1*cy3-cy1*cy2*wy)*cy4+cy2*cy3*sampling
                    - cy1*cy3+cy1*cy2*wy+cy2*cy2*sy)/(cy2*cy2);
  data[point][6]  = rpull*(wy+(cy1*cy3-cy1*cy2*wy)*(1-cy4)/(cy1*cy2));
             /*  End modely */
}
                  /*         end model.c       */

                 /*  pmenu.c  21:00:25  11/13/1987 */

#include "stdio.h"
#define bell '\007'

path_edit(datapath)
float datapath[][6];

 {
extern int feof();
extern FILE *fopen();
extern int  fclose();
FILE  *prt;
extern key_getc();
int i,j,k,l,press,check,ii,jj,endf;
char  pname[15];
   check = 0;
   k = 1;
   i = datapath[0][0]+1;
   press = 4209;
   do
    {
```

```
switch(press)
  {
    case 11875  :  /*  c clear data */
      i = 1;
      break;
    case 15104  :  /*  F1  */
      crt_cls();
      printf("Type Exit and Press Return to go Back");
      system("a:command");
      break;
    case 15360  :  /*  F2  save path file  */
      crt_cls();
      block1(0,0,23,79);
      crt_srcp(12,10,0);
      printf("Path File Name to save:");
      scanf("%s",pname);
      crt_srcp(14,10,0);
      printf("Save path :");
      prt = fopen(pname,"w");
      for(ii=0;ii<i;ii++)
       {
        for(jj=0;jj<=5;jj++)
         fprintf(prt,"%f  ",datapath[ii][jj]);
        fprintf(prt,"\n");
        crt_srcp(14,22,0);
        printf("%d",ii);
       }
      fclose(prt);
      break;
    case 15616  :  /*  F3  read path file  */
      crt_cls();
```

```
    block1(0,0,23,79);
    crt_srcp(12,10,0);
    printf("Path File Name to read:");
    scanf("%s",pname);
    crt_srcp(14,10,0);
    printf("Read path :");
    prt = fopen(pname,"r");
    ii = -1;
    do
     {
       ii++;
       for(jj=0;jj<=5;jj++)
       fscanf(prt,"%f ",&datapath[ii][jj]);
       crt_srcp(14,22,0);
       printf("%d",ii);
       endf = feof(prt);
     }while(!endf); /* not end of file (zero) */
    fclose(prt);
    i = ii;
    break;
  case 7777  :
    datapath[i][0] = 1;
    crt_cls();
    crt_srcp(3,10,0);
    printf(" Arc Path no. %d ",i);
    crt_srcp(5,15,0);
    printf(" Center  point X & Y is ");
    scanf("%f %f \n",&datapath[i][1],&datapath[i][2]);
    crt_srcp(7,15,0);
    printf(" Radius of circle is ");
    scanf("%f\n",&datapath[i][3]);
```

```
        crt_srcp(9,15,0);
        printf(" start angle is ");
        scanf("%f\n",&datapath[i][4]);
        crt_srcp(11,15,0);
        printf(" End angle is ");
        scanf("%f\n",&datapath[i][5]);
        for (l=1;l<=3;l++)
         datapath[i][l] = datapath[i][l]/1000;
        i = k+1;
        break;
    case 9836 :
        datapath[i][0] = 2;
        crt_cls();
        crt_srcp(3,10,0);
        printf(" Line Path no. %d ",i);
        crt_srcp(5,15,0);
        printf(" Start point X & Y is ");
        scanf("%f %f \n",&datapath[i][1],&datapath[i][2]);
        crt_srcp(7,15,0);
        printf(" End   point X & Y is ");
        scanf("%f %f \n",&datapath[i][3],&datapath[i][4]);
        datapath[i][5] = 0;
        for (l=1;l<=4;l++)
         datapath[i][l] = datapath[i][l]/1000;
        i = k+1;
        break;
    case 4209 :
        crt_cls();
        break;
    default :
        while(check != 283)
```

```
            {
             putchar(bell);
             crt_srcp(24,15,0);
             printf(" Select  Error  Press Esc to continue     ");
             check = key_getc();
            }
       }
    datapath[0][0] = i-1;
    crt_cls();
    printf("                              Path input Data ");
    for(j=1;j<i;j++)
     {
    check = datapath[j][0];
     switch(check)
       {
         case 1  :
           crt_srcp(j*2,5,0);
           printf(" Arc  no.  Xcenter   Ycenter    Radius
                    Start Angle  End Angle\n");
          printf("            %d      %5.3f      %5.3f      %5.3f
          %6.2f       %6.2f ",j,datapath[j][1],datapath[j][2],
          datapath[j][3],datapath[j][4],datapath[j][5]);
           break;
        case 2 :
           crt_srcp(j*2,5,0);
printf(" Line no.  X start   Y start     X end       Y end \n ");
printf("          %d      %5.3f      %5.3f      %5.3f      %5.3f",
 j,datapath[j][1],datapath[j][2],datapath[j][3],datapath[j][4]);
           break;
       }
  }
```

```
crt_srcp(22,10,0);
 printf("------------------------------------------------------");
    crt_srcp(24,12,0);
 printf("Arc, Line, Edit, Clear, F1 Dos, F2 Save, F3 Read, Quit");
    press = key_getc();
    k = i;
    if(press == 4709)
      {
        if(i != 1)
         {
          k = k-1;
          crt_srcp(24,12,0);
         printf("                                                ");
          crt_srcp(24,15,0);
          printf(" Edit Path no.   ");
          i = key_getc();
          j = i >> 7;
          j = j << 7;
          i = i - j - 48;
          if (i<=k && i != 0)
           {
            crt_srcp(24,15,0);
            printf(" Select Arc  or  Line ");
            press = key_getc();
             if(press == 4209)
              {
               press = 0;
               i     = k + 1;
              }
           }
            else
```

```
                    i = k +1;
                }
            }
        check = 0;
      }while(press != 4209 && press != 4177);
  }

                    /*  end pmenu.c  */

                /*  calpath.c  9:11:27  11/14/1987  */

#include "stdio.h"
#define Maxp 1280
#define exce 50

calpath(pvp,sector,numpoint,datapath)
float sector,datapath[][6];
int *numpoint,pvp;

 {
extern float data[][10];
extern key_getc();
int i,j,k,start,endpoint,check,endp;
float data_select[6];
     start = 0;
     endpoint = 0;
     endp  = datapath[0][0];
     for(i=1;i<=endp;i++)
      {
        for(j=1;j<=5;j++)
          {
```

```
            data_select[j] = datapath[i][j];
          }
          if (datapath[i][0] == 2)
            {
              line(sector,data_select,start,&endpoint);
              if (endpoint > Maxp) i = endp;
            }
             else
             {
               arc(sector,data_select,start,&endpoint);
               if (endpoint > Maxp) i = endp;
             }
        start = endpoint;
      }
      *numpoint         = endpoint;
}

                    /*  end calpath.c  */

               /*  path.c    13:04:45  10/29/1987 */

                    /*  arc   */
#define pi 3.1415926536
arc(sector,selct,start,end)
float sector,selct[];
int    start,*end;
{
 extern double sin(),cos(),acos();
 extern  float  data[][10];
 int    i,check,stop;
 float  num,angle,seta;
```

```
angle  = acos((2*selct[3]*selct[3]-sector*sector)/
              (2*selct[3]*selct[3]));
num    = (selct[5] - selct[4])*pi/(180*angle);
check  = num;
stop   = start + check;
if (stop > Maxp) *end = Maxp + exce;
 else
{
seta = selct[4]*pi/180;
for(i=start;i<=stop;i++)
  {
  data[i][1] = selct[3]*cos(seta) + selct[1];
  data[i][2] = selct[3]*sin(seta) + selct[2];
  seta += angle;
  }
if ((num - check) > 0.5)
  {
    data[stop+1][1] = selct[3]*cos(selct[5]*pi/180) + selct[1];
    data[stop+1][2] = selct[3]*sin(selct[5]*pi/180) + selct[2];
    *end      = stop+1;
  }
   else
    {
      data[stop][1]   = selct[3]*cos(selct[5]*pi/180) + selct[1];
      data[stop][2]   = selct[3]*sin(selct[5]*pi/180) + selct[2];
      *end      = stop;
    }
}
}

                /*   End arc    */
```

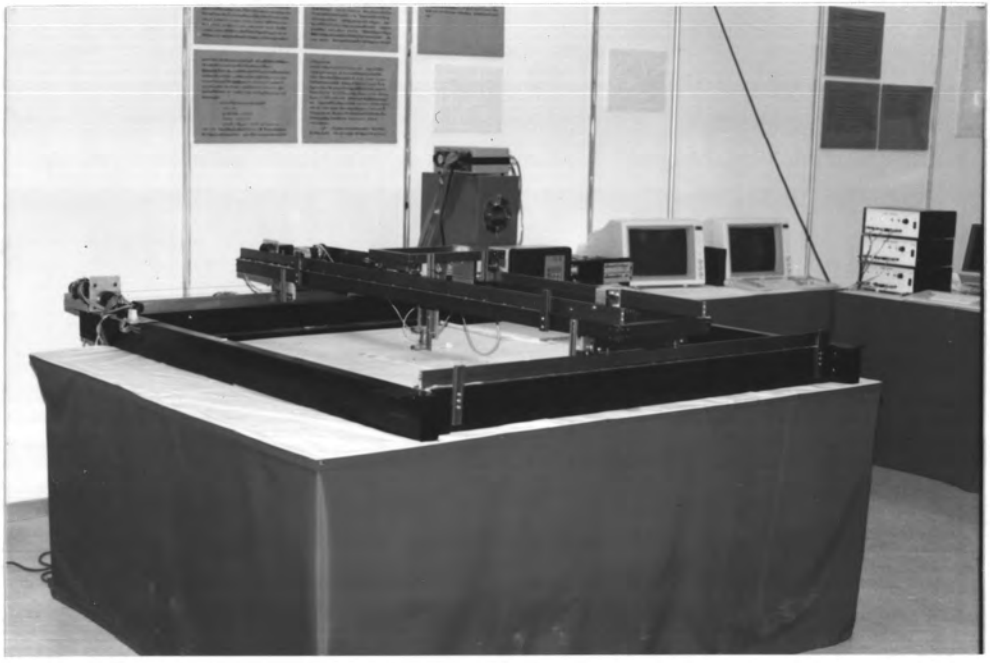
```
                    /*   Start Line */

line(sector,selct,start,end)
float  sector,selct[];
int    start,*end;
{
 extern double sqrt(),sin(),cos(),atan2();
 extern float  data[][10];
 int    i,check,stop;
 float  num,xunit,yunit,angle,lenght,subl;
   angle  = atan2((selct[4]-selct[2]),(selct[3]-selct[1]));
   xunit  = cos(angle);
   yunit  = sin(angle);
   lenght = sqrt((selct[3]-selct[1])*(selct[3]-selct[1])
                 + (selct[4]-selct[2])*(selct[4]-selct[2]));
   num    = lenght/sector;
   check  = num;
   stop   = start + check;
   if (stop > Maxp) *end = Maxp + exce;
    else
   {
   subl   = 0;
   for(i=start;i<=stop;i++)
     {
     data[i][1]  = subl*xunit + selct[1];
     data[i][2]  = subl*yunit + selct[2];
     subl += sector;
     }
   if ((num - check) > 0.5)
     {
       data[stop+1][1] = lenght*xunit + selct[1];
```

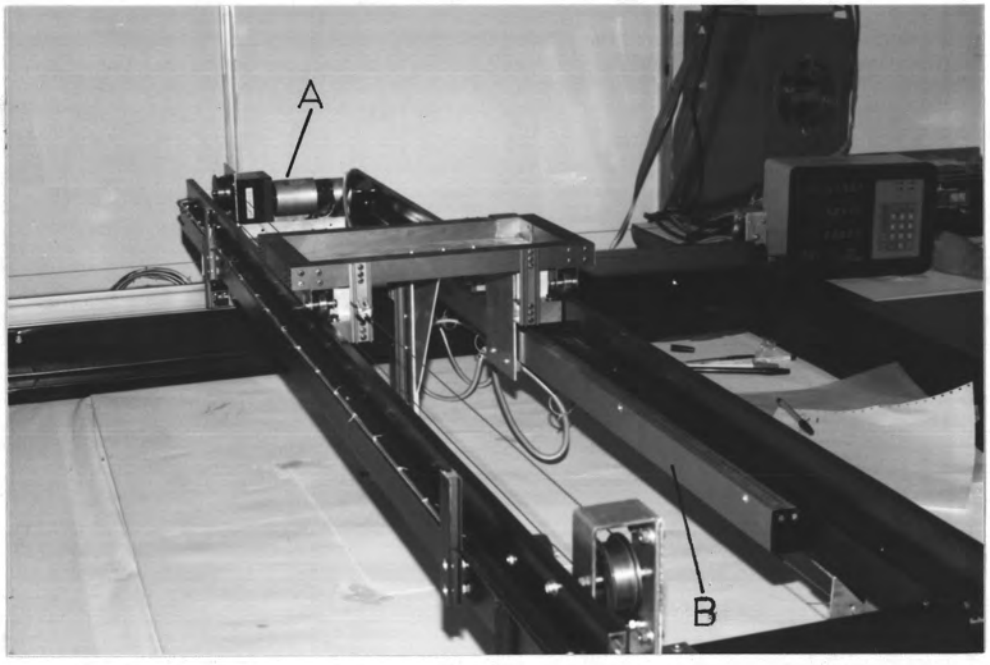
```
            data[stop+1][2] = lenght*yunit + selct[2];
            *end       = stop+1;
          }
          else
          {
            data[stop][1]   = lenght*xunit + selct[1];
            data[stop][2]   = lenght*yunit + selct[2];
            *end       = stop;
          }
        }
      }

                        /*  End  Line  */
                        /*  end  path.c  */
```

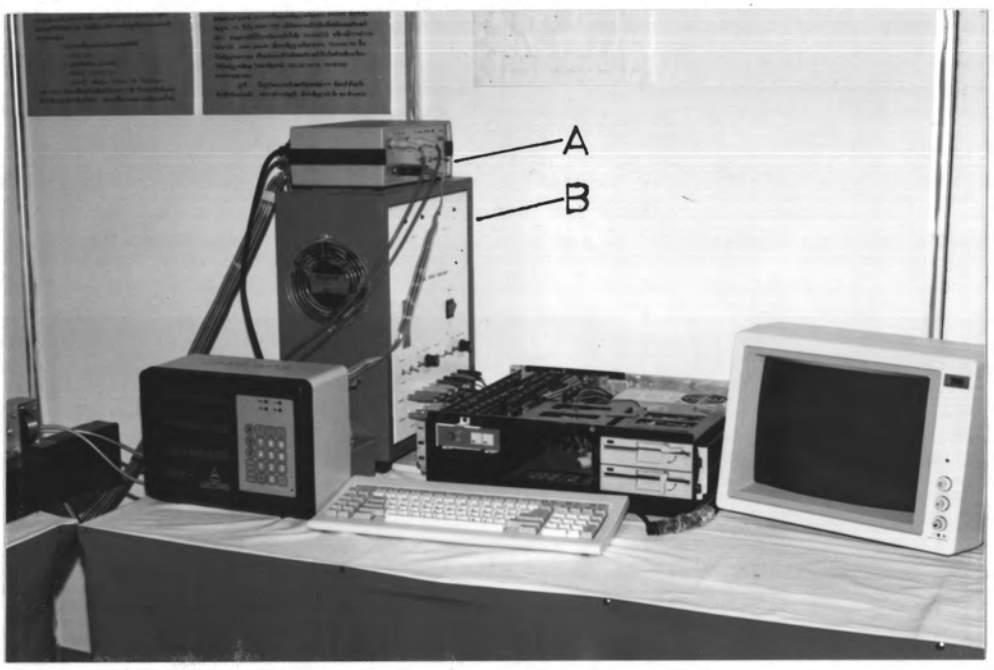
ภาคผนวก จ.
รูปแสดงอุปกรณ์ที่ใช้ในการทดลอง



โต๊ะตัดแผ่นเหล็กที่พัฒนาขึ้น

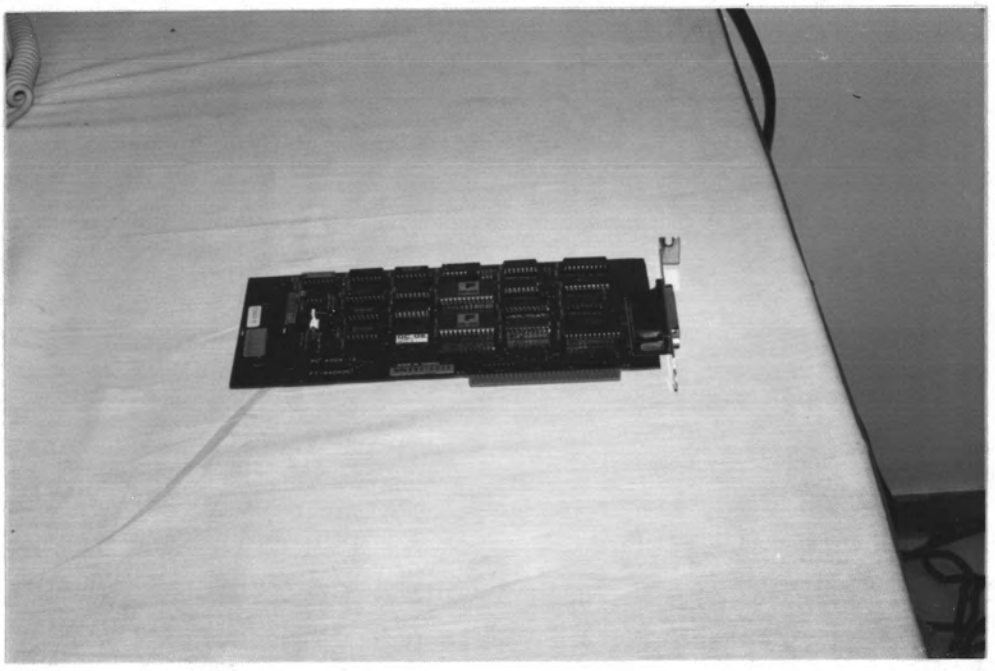


A - มอเตอร์และระบบล้อสายพาน, B - ออปติคอลลีเนียร์เอ็นโคดเดอร์

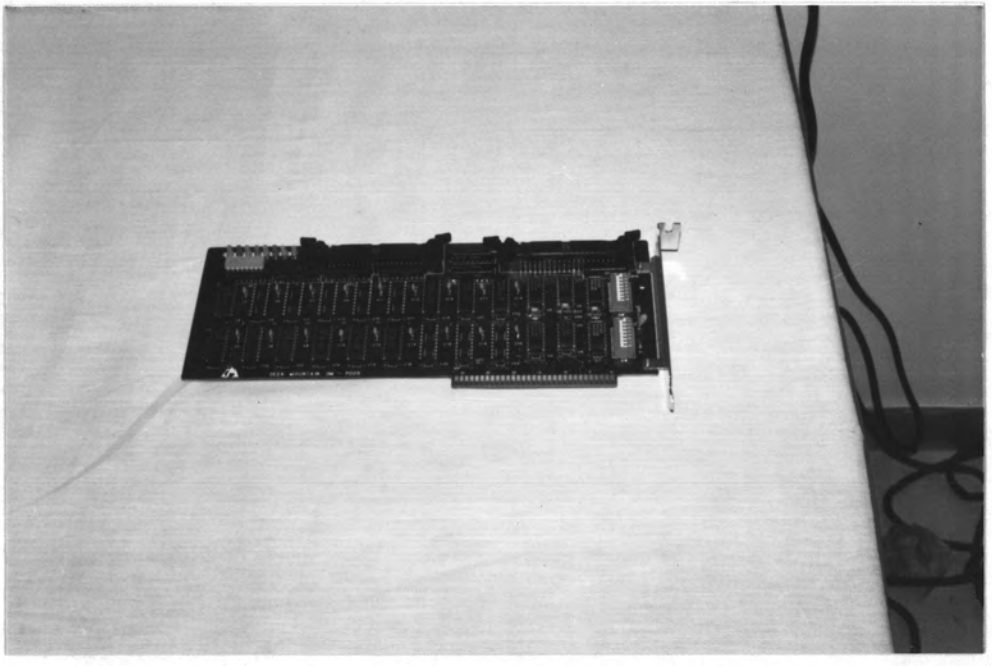


A – ชุดดีโคดเดอร์และวงจรนับ, B – เพาเวอร์แอมปลิไฟเออร์

ตัวแยกสัญญาณและรักษาระดับแรงดัน ตัวกรองความถี่สูง



ดิจิตอลอนาลอคคาร์ด



ที.ที.แอล.คาร์ด

ประวัติผู้เขียน

นาย วรงค์สิทธิ มารัตน์ เกิดเมื่อวันที่ 3 สิงหาคม พ.ศ. 2506 ที่อำเภอเมือง จังหวัดอุบลราชธานี สำเร็จการศึกษาขั้นปริญญาตรีจาก มหาวิทยาลัยขอนแก่น สาขาวิชาวิศวกรรมเครื่องกล เมื่อปีการศึกษา 2528