

## อุปกรณ์และวิธีการ

### อุปกรณ์

จากวัตถุประสงค์ของงานเราได้ออกแบบวิธีการศึกษาการระบุเอกลักษณ์โดยใช้ข้อมูลจากการทดลองของระบบมอเตอร์ไฟฟ้ากระแสตรงของจริง ซึ่งมีอุปกรณ์ดังนี้

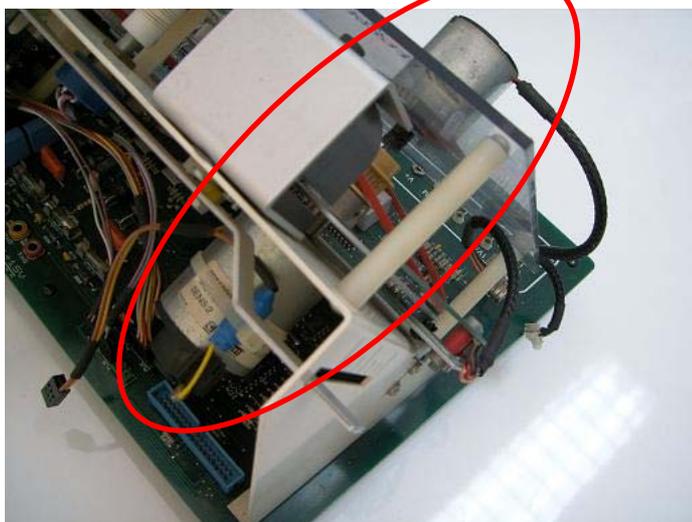
1. คอมพิวเตอร์ที่มีช่องเสียบการ์ดชนิด PCI และ โปรแกรม MATLAB ที่มี System Identification Toolbox 1 ชุด ในที่นี้เราจะใช้โปรแกรม MATLAB เวอร์ชัน 7.0.4.365 (R14) Service Pack 2 แสดงตามภาพที่ 32 เราใช้คอมพิวเตอร์ที่สามารถต่อใช้งานร่วมกันกับการ์ดสำหรับแปลงสัญญาณจาก Analog เป็น Digital เพื่อเก็บข้อมูล ในที่นี้จะใช้การ์ด PCI-9112 ซึ่งจะติดต่อกับคอมพิวเตอร์โดยใช้ช่องเสียบการ์ดชนิด PCI และเราจะใช้โปรแกรม MATLAB ในคอมพิวเตอร์ช่วยในการคำนวณและประมวลผล



ภาพที่ 32 คอมพิวเตอร์และ โปรแกรม MATLAB 7.0.4.365 (R14) Service Pack 2

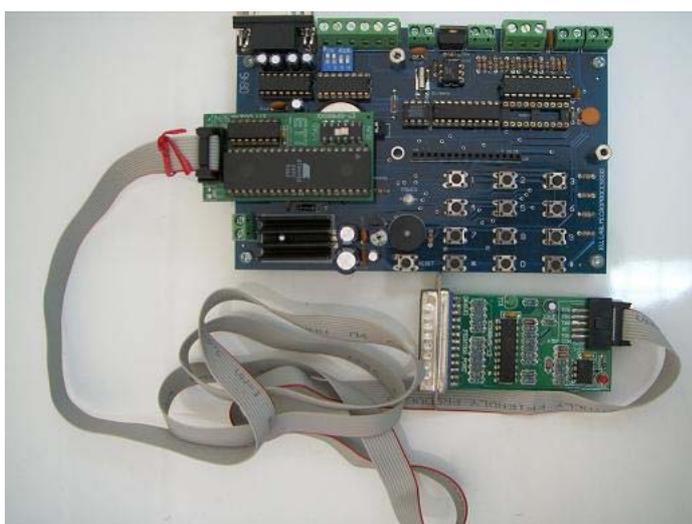
2. DC Servo Motor ต่อเชื่อมกับ Tacho-Generator 1 ชุด แสดงตามภาพที่ 33 เราได้เลือกมอเตอร์ไฟฟ้ากระแสตรงจากชุดทดลองรุ่น MECHANICAL UNIT Control & Instrumentation 33-100 ซึ่งเป็นชุดทดลองที่ใช้งานในห้องปฏิบัติการควบคุมซึ่งติดตั้งอยู่กับชุดทดลองการควบคุมการเคลื่อนที่เพราะมีการติดตั้งมอเตอร์เข้ากับ Tacho-Generator ก่อนข้างเรียบร้อยแล้ว ทำการตัดแปลง

ไม่มากก็สามารถใช้ทำการทดลองได้ แต่ก็ยังมีโหนดบางส่วนที่ไม่สามารถถอดออกจากตัวมอเตอร์ได้ทั้งหมดซึ่งก็อาจจะมีผลกระทบกับการทดลองบ้างเล็กน้อย มอเตอร์ไฟฟ้ากระแสตรงดังกล่าวเป็นมอเตอร์ 24VDC และมี Tacho-Generator 24V / 5000RPM



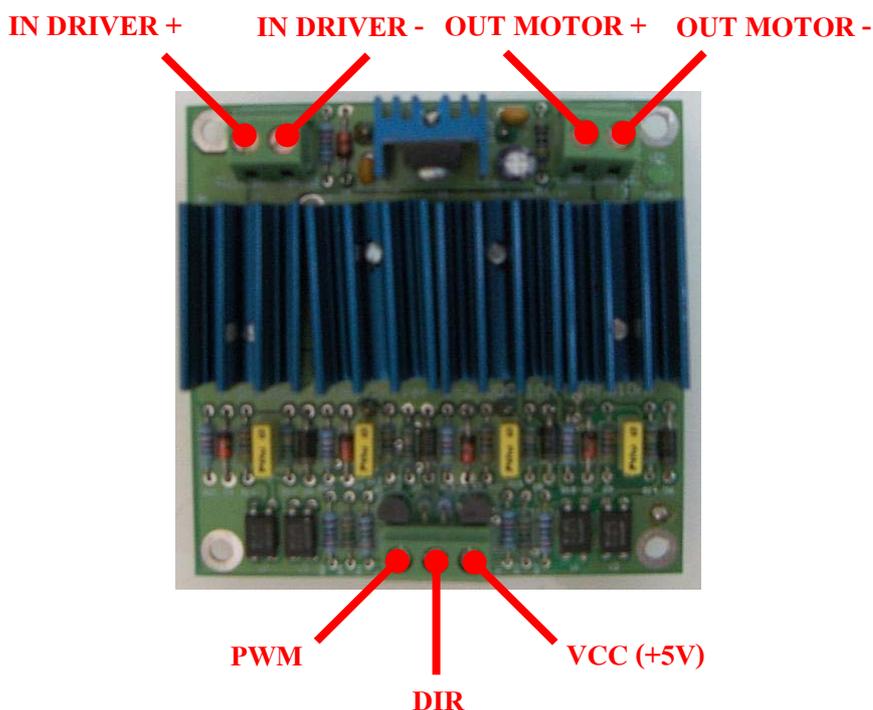
ภาพที่ 33 DC Servo Motor ต่อเชื่อมกับ Tacho-Generator ในชุดทดลองการควบคุมการเคลื่อนที่

3. บอร์ดไมโครคอนโทรลเลอร์เนกประสงค์พร้อมชุดดาวน์โหลดโปรแกรม 1 ชุด แสดงตามภาพที่ 34 บอร์ดดังกล่าวทำหน้าที่สร้างสัญญาณต่างๆ ให้กับระบบที่เราจะทำการทดสอบ



ภาพที่ 34 บอร์ดไมโครคอนโทรลเลอร์เนกประสงค์พร้อมชุดดาวน์โหลดโปรแกรม

4. บอร์ด Driver 1 ชุด แสดงตามภาพที่ 35 บอร์ดดังกล่าวทำหน้าที่แยกกราวด์ระหว่าง สัญญาณคำสั่งและสัญญาณที่ใช้ขับมอเตอร์และปรับเปลี่ยน Amplitude ของสัญญาณคำสั่ง (แต่ไม่ได้เปลี่ยนลักษณะของสัญญาณ) ให้มีแรงดันและกระแสที่สูงพอที่จะสามารถนำไปป้อนให้กับ ระบบ เราได้เลือกใช้บอร์ดของบริษัท Inex ซึ่งสามารถจ่ายแรงดันให้มอเตอร์ได้ 24VDC รับกระแส สูงสุดต่อเนื่องได้ 10A



ภาพที่ 35 บอร์ด Driver ผลิตโดยบริษัท Inex

5. การ์ดสำหรับแปลงข้อมูลจาก Analog เป็น Digital (A/D) 1 ชุด แสดงตามภาพที่ 36 ในที่ เราได้เลือกการ์ดรุ่น PCI-9112 มาทำหน้าที่เก็บข้อมูลการทดลองเพื่อที่จะนำไปใช้ในการ ประมวลผลผลการระบุเอกลักษณ์ ซึ่งมีความละเอียดในการแปลงข้อมูลจาก Analog เป็น Digital ได้ ละเอียด 12 bits มีช่องรับสัญญาณ 8 ช่องสัญญาณ และมีอัตราการสุ่มเก็บสัญญาณได้สูงสุด 110 KHz (เมื่อใช้เพียง 1 ช่องสัญญาณ) การ์ดดังกล่าว 1 ชุดจะประกอบไปด้วยตัวการ์ดที่สามารถ ต่อเชื่อมกับคอมพิวเตอร์ที่มีช่องเสียบแบบ PCI เทอร์มินอล สายต่อเชื่อมระหว่างตัวการ์ดกับ เทอร์มินอลและซีดีรอมข้อมูลการใช้งานพร้อมทั้งตัวอย่าง โปรแกรม MATLAB ที่ใช้ติดต่อกับการ์ด



(1) การ์ด ADC PCI-9112



(2) แสดงสายสัญญาณ DB-37 และเทอร์มินอล รุ่น DIN-37D

ภาพที่ 36 การ์ดสำหรับแปลงข้อมูลจาก Analog เป็น Digital และอุปกรณ์เพิ่มเติม

6. Power supply 2 ชุด แสดงตามภาพที่ 37 ใช้เป็นแหล่งจ่ายไฟฟ้าให้กับอุปกรณ์การทดลอง ในการทดลองใส่อินพุตเป็นชนิด Step ให้กับระบบ เราจะใช้ Power supply ผลิตโดยบริษัท inSTEK เป็นแหล่งจ่ายไฟขนาด 18V เพื่อใช้ในการสร้างอินพุตชนิด Step ในการทดลองใส่อินพุตเป็นชนิด PRBS เราจะใช้ Power supply inSTEK เป็นแหล่งจ่ายไฟขนาด 9-12V ให้กับบอร์ดไมโครคอนโทรลเลอร์เอนกประสงค์ และจะใช้ Power supply ชนิดถ่านชาร์จ Ni-MH ผลิตโดยบริษัท GP Batteries เป็นแหล่งจ่ายไฟขนาด 18V ให้กับ Driver Board เพื่อนำไปใช้สร้างสัญญาณเพื่อขับมอเตอร์ เหตุผลที่เราต้องนำถ่านชาร์จมาเป็นแหล่งจ่ายไฟให้กับ Driver Board นั้นเพราะว่า ถ้าเรา Power supply inSTEK ไปเป็นแหล่งจ่ายไฟสำหรับการสร้างสัญญาณ PRBS Power supply จะขึ้นเตือนกระแสเกิน (Over Current) เนื่องจากไม่สามารถจ่ายกระแสได้มากพอต่อความต้องการของ



(1) inSTEK Model SPS-3610



(2) GP 1800mAh AA Ni-MH Batteries

### ภาพที่ 37 Power supply ที่ใช้ในการทดลอง

มอเตอร์ขณะเกิดการกระชากของกระแสได้ เราจึงเลือกใช้ถ่านชาร์จมาเป็นแหล่งจ่ายไฟแทน จากการทดลองสร้างสัญญาณอินพุตเป็นชนิด PRBS ให้กับระบบมอเตอร์โดยใช้แหล่งจ่ายไฟเข้า Driver Board เป็นถ่านชาร์จ พบว่าขณะเกิดการกระชากของกระแส สัญญาณแรงดันที่ขั้วมอเตอร์ที่วัดได้จาก Oscilloscope ยังคงมีลักษณะใกล้เคียงกับสัญญาณ PRBS อยู่แสดงตามภาพที่ 51 (ก) เราจึงนำถ่านชาร์จมาเป็นแหล่งจ่ายไฟให้กับ Driver Board เพื่อสร้างสัญญาณ PRBS ให้กับระบบมอเตอร์ได้

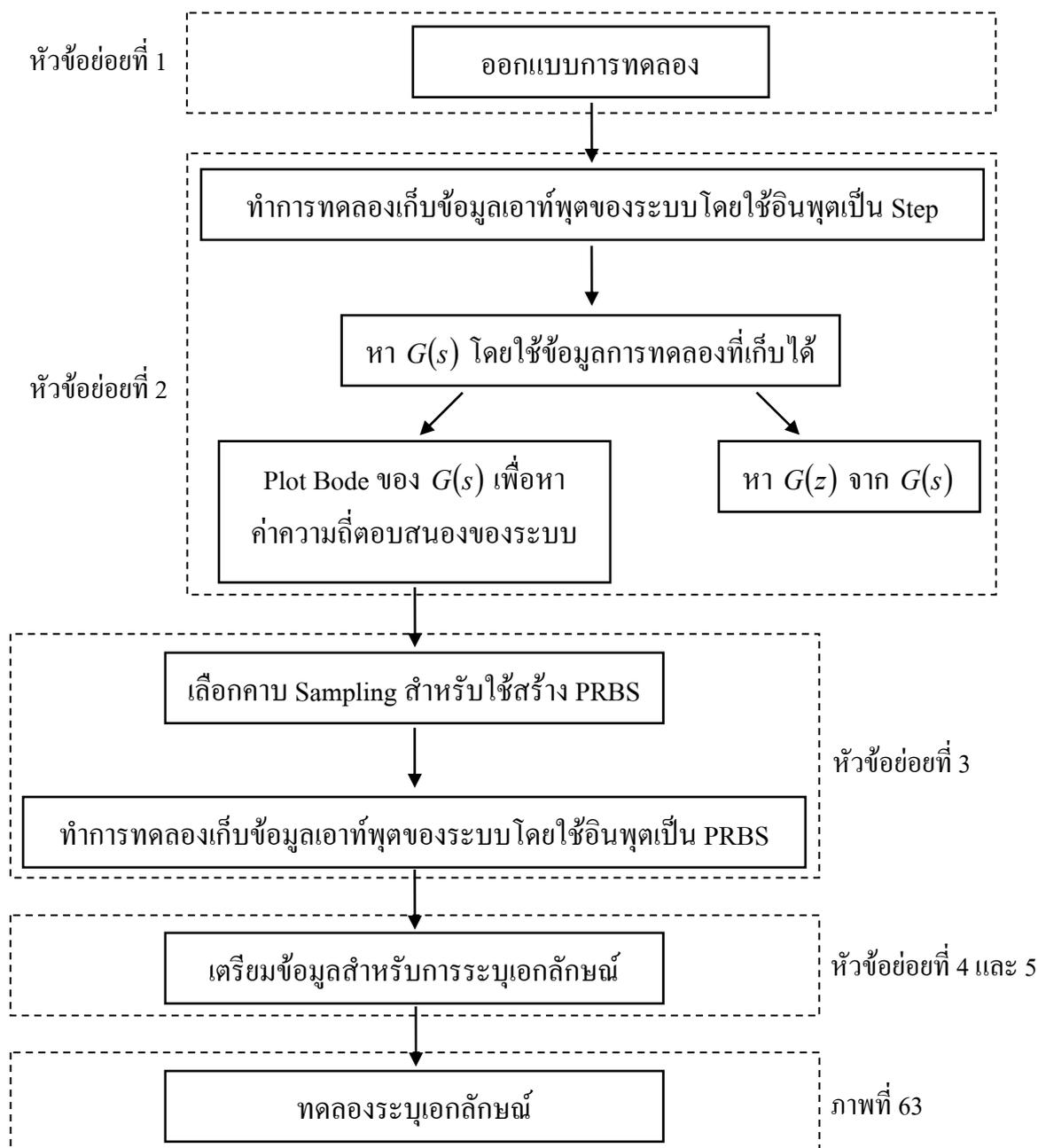
7. Oscilloscope 1 ชุด แสดงตามภาพที่ 38 เราใช้เป็นเครื่องมือวัดและตรวจสอบสัญญาณการทำงานของอุปกรณ์ต่างๆ ในการทดลอง



ภาพที่ 38 Oscilloscope

### วิธีการ

ลำดับขั้นตอนการทดลองสำหรับการจัดเก็บและเตรียมข้อมูลเพื่อนำไปใช้ในการระบุเอกลักษณ์สามารถเขียนเป็นแผนผังได้ตามภาพที่ 39

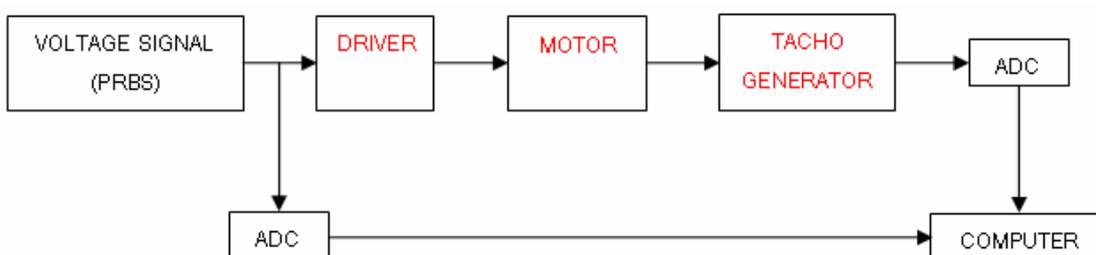


ภาพที่ 39 แสดงแผนภาพลำดับขั้นตอนการทดลอง

## 1. ออกแบบการทดลอง

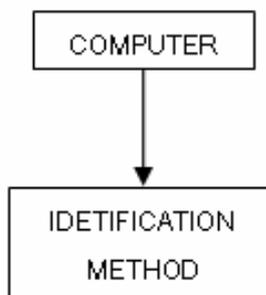
ในขั้นตอนการออกแบบการทดลองเราได้ออกแบบการทดลองโดยแบ่งเป็น 2 ส่วน คือ ส่วนสำหรับเก็บข้อมูลการทดลองและส่วนสำหรับประมวลผลข้อมูลการทดลองเพื่อที่จะระบุเอกลักษณ์ของระบบมอเตอร์ไฟฟ้ากระแสตรงได้

1. สร้างชุดทดลองสำหรับการเก็บข้อมูลการระบุเอกลักษณ์ของระบบมอเตอร์ไฟฟ้ากระแสตรง แสดงดังภาพที่ 40



ภาพที่ 40 Block Diagram แสดงการทดลองเพื่อเก็บข้อมูล

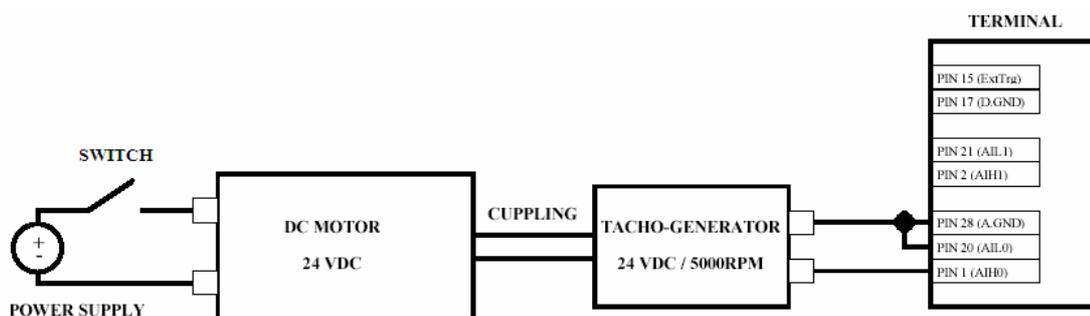
2. หลังจากเราได้ข้อมูลจากการทดลองจากข้อที่ 1. เราจะใช้ข้อมูลดังกล่าวในการศึกษาและเปรียบเทียบผลการทดสอบทฤษฎีการระบุเอกลักษณ์ของระบบมอเตอร์ไฟฟ้ากระแสตรงหลายๆ ทฤษฎี ซึ่งจะใช้โปรแกรม MATLAB ในคอมพิวเตอร์เป็นตัวช่วยในการคำนวณและประมวลผลในการหาแบบจำลองทางคณิตศาสตร์และเปรียบเทียบผลการทดลอง แสดงดังภาพที่ 41



ภาพที่ 41 Block Diagram แสดงการทดลองเพื่อประมวลผลข้อมูล

## 2. การทดลองหาค่าความถี่ตอบสนองของมอเตอร์โดยใช้สัญญาณอินพุตเป็น Step

เริ่มต้นที่การทดลองหาค่าความถี่ตอบสนองของมอเตอร์โดยใช้สัญญาณอินพุตเป็น Step ขนาด 18V โดยทำการจัดแบบไฟของวงจรการทดลองดังภาพที่ 42



ภาพที่ 42 แบบวงจรของการทดลองหาค่าความถี่ตอบสนองของมอเตอร์โดยใช้สัญญาณอินพุตเป็น Step

ในการทดลองเราใช้ POWER SUPPLY inSTEK โดยปรับแรงดันขนาด 18V และปรับกระแสให้พอกับความต้องการของระบบขณะที่ใส่สัญญาณอินพุตแบบ Step นั่นคือเมื่อปิดวงจรโดยใช้ SWITCH ในภาพที่ 42 POWER SUPPLY จะต้องไม่ขึ้นเตือนกระแสเกิน (Over Current) ภายหลังจากใส่สัญญาณอินพุตดังกล่าว โรเตอร์ของมอเตอร์จะเกิดการเคลื่อนที่แบบหมุน การเคลื่อนที่ที่จะถูกถ่ายทอดไปยังแกนโรเตอร์ของ Tacho-Generator โดยอุปกรณ์ที่เรียกว่า Cuppling ทำให้โรเตอร์ของ Tacho-Generator เกิดการเคลื่อนที่ตามกันไปกับโรเตอร์ของมอเตอร์ การหมุนของโรเตอร์ของ Tacho-Generator จะทำให้เกิดความต่างศักย์ขึ้นที่ขั้วของ Tacho-Generator ความต่างศักย์ที่เกิดขึ้นจะมีความสัมพันธ์กับความเร็วการหมุนของโรเตอร์ของ Tacho-Generator โดยจะมีความสัมพันธ์กันเป็นเชิงเส้น ในที่นี้คือ 24V / 5000RPM ทำให้เราสามารถวัดความเร็วการหมุนของมอเตอร์ได้โดยการวัดความต่างศักย์ที่ขั้วของ Tacho-Generator เราทำการเก็บค่าความต่างศักย์โดยใช้การ์ด PCI-9112 ทำได้โดยการต่อขั้วของ Tacho-Generator เข้ากับช่องสัญญาณที่ 0 ของการ์ด PCI-9112 แสดงตามภาพที่ 42 จากนั้นใช้โปรแกรม MATLAB ชื่อ Syn\_Muti\_V\_Edit.m แสดงในภาคผนวก ง โดยทำการกำหนดค่าของตัวแปรต่างๆดังนี้

- TrigSource = 1      คือ กำหนดให้ใช้ Clock ภายในเป็นตัวสร้างความถี่ในการ

Sampling

- sample\_rate = 2000      คือ กำหนดค่าความถี่ Sampling ทุกช่องรวมกันเท่ากับ 2000Hz
- ai\_buf = 40960          คือ กำหนดค่าของพื้นที่ๆใช้เก็บข้อมูล
- data\_size = 20480      คือ กำหนดค่าของพื้นที่ๆใช้เก็บข้อมูล

ทำการ Run โปรแกรม ผลที่ได้คือ การ์ด PCI-9112 จะทำการเก็บข้อมูลสัญญาณที่ช่องสัญญาณที่ 0 และช่องสัญญาณที่ 1 ด้วยความถี่ Sampling ช่องสัญญาณละ 1000 Hz ข้อมูลของช่องสัญญาณที่ 0 จะถูกเก็บไว้ในตัวแปร signal\_ch0 และข้อมูลของช่องสัญญาณที่ 1 จะถูกเก็บไว้ในตัวแปร signal\_ch1 จากแบบไปวงจรรูปที่ 42 จะเห็นว่าในการทดลองนี้เราเก็บข้อมูลเฉพาะช่องสัญญาณที่ 0 จึงจะไม่กล่าวถึงข้อมูลในตัวแปร signal\_ch1 และส่วนของโปรแกรม Syn\_Muti\_V\_Edit.m อื่นๆ ที่กระทำกับข้อมูลของ signal\_ch1

ขณะนี้เราได้ผลตอบสนองของระบบโดยใช้สัญญาณอินพุตเป็น Step อยู่ที่ตัวแปร signal\_ch0 เรียบร้อยแล้ว เพื่อความเข้าใจง่ายเราจึงทำการเปลี่ยนตั้งชื่อตัวแปรใหม่โดยใช้คำสั่งโปรแกรม MATLAB ดังนี้

```
>> signal1_raw_step = signal_ch0;
```

และเราสามารถเก็บข้อมูลการทดลองแบบถาวรได้โดยเก็บข้อมูลดังกล่าวลงใน Hardisk และเพื่อความเข้าใจถึงที่มาของข้อมูล ในการเก็บข้อมูลเราจึงตั้งชื่อข้อมูลดังกล่าวเป็นชื่อเดียวกันกับชื่อการทดลอง ซึ่งสามารถทำได้โดยใช้คำสั่งโปรแกรม MATLAB ดังนี้

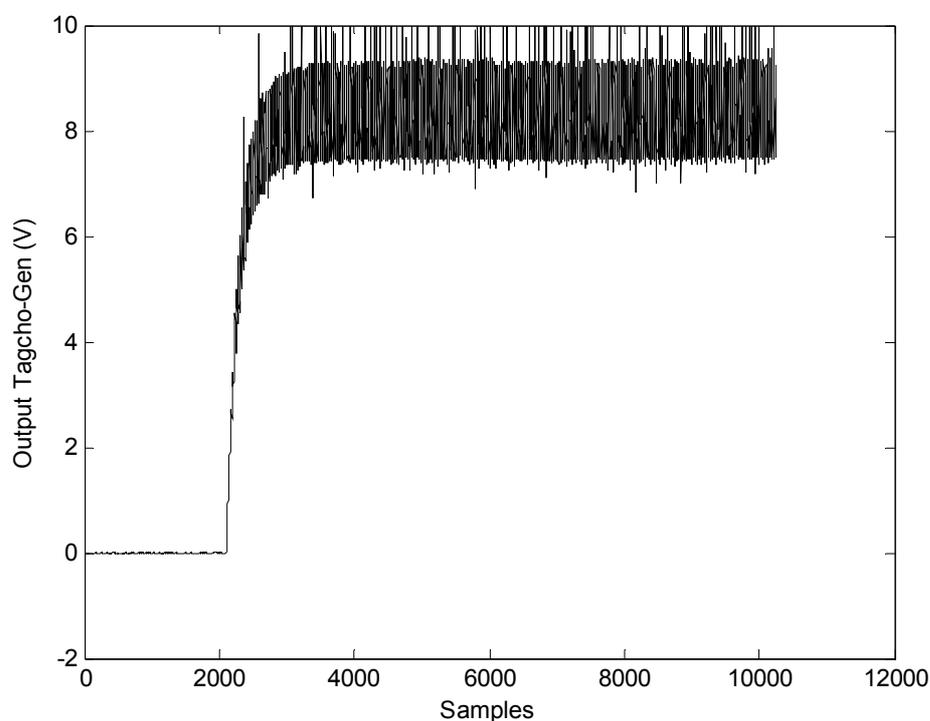
```
>> save ('signal1_raw_step', 'signal1_raw_step');
```

ตัวแปร signal1\_raw\_step จะถูกเก็บในชื่อของ signal1\_raw\_step.mat ในบริเวณ Current Directory ที่โปรแกรม MATLAB ใช้งานอยู่ในขณะนั้น

ณ ขณะนี้เราได้ทำการทดลองเพื่อเก็บข้อมูลเรียบร้อยแล้ว ต่อไปจะทำการวิเคราะห์ข้อมูลดังกล่าวได้โดยคำสั่งโปรแกรม MATLAB ดังนี้ กำหนด Current Directory ของโปรแกรม MATLAB ไปอยู่ที่เดียวกันกับที่เก็บข้อมูล signal1\_raw\_step.mat จากนั้นพิมพ์คำสั่ง

```
>> load ('signal1_raw_step');
>> plot (signal1_raw_step, 'black')
>> xlabel ('Samples')
>> ylabel ('Output Tagcho-Gen (V)')
```

ทำให้เราได้ผลตอบสนองที่วัดได้จาก Tacho-Generator แสดงตามภาพที่ 43



ภาพที่ 43 กราฟ Step Response (Sampling Rate 1 KHz)

ข้อสำคัญของการทดลองนี้คือ ขณะที่ใส่สัญญาณอินพุตที่เป็น Step ให้กับมอเตอร์จะเกิดการกระชากของกระแสไฟที่มากกว่าปกติเข้าสู่มอเตอร์ แหล่งจ่ายไฟที่นำมาใช้ก็ควรจะมีความสามารถในการจ่ายกระแสได้พอกับความถี่กระชาก

จากภาพที่ 43 กราฟของผลตอบสนองมีสัญญาณรบกวนเข้ามาด้วย เพื่อให้ได้สัญญาณที่มีคุณภาพมากขึ้น จึงควรใช้ Filter กรองสัญญาณรบกวนออก เนื่องจากสัญญาณรบกวนที่เห็นแบบ Spike Noise และ Noise ทั่วไป จึงนำ Median Filter มาใช้งาน และลองนำ Moving Average Filter เข้ามาใช้ด้วยเพื่อจะได้ทำการเปรียบเทียบได้ประสิทธิภาพของ Filter แต่ละชนิดได้ การใช้ Filter ทั้ง

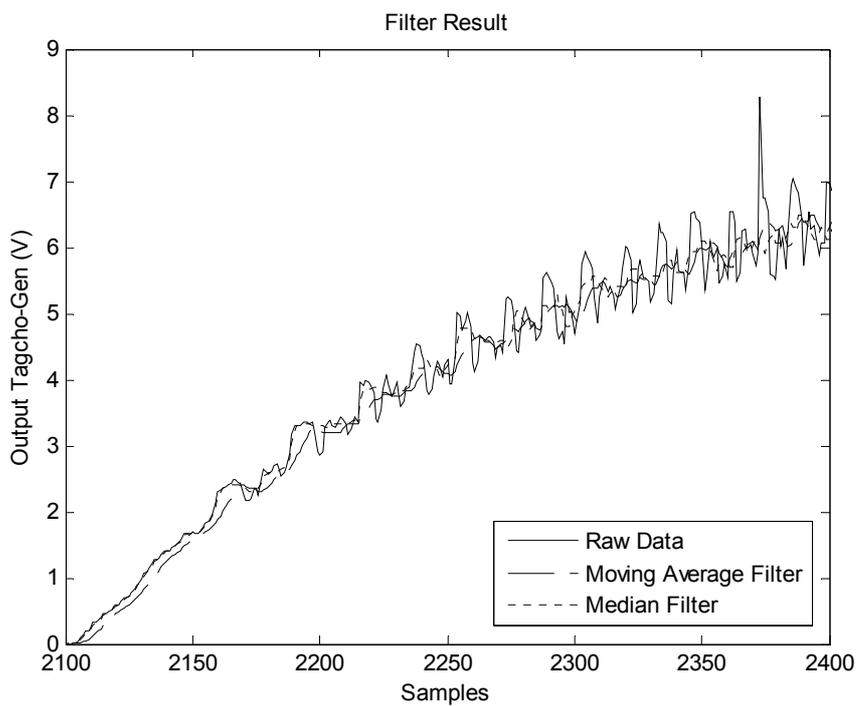
2 ชนิดโดยกำหนดให้ความกว้างของ Window เท่ากับ 10 (ถ้าความกว้างของ Window มีค่าน้อยเกินไปเราจะพบว่า Filter จะไม่สามารถกรองสัญญาณรบกวนออกไปได้) สามารถทำได้ใช้คำสั่งโปรแกรม MATLAB ดังนี้

```
>> a = 1;
>> b = (1/10)*[1 1 1 1 1 1 1 1 1 1];
>> mov_s1 = filter (b, a, signal1_raw_step);
>> med_s1 = medfilt1 (signal1_raw_step, [10]);
```

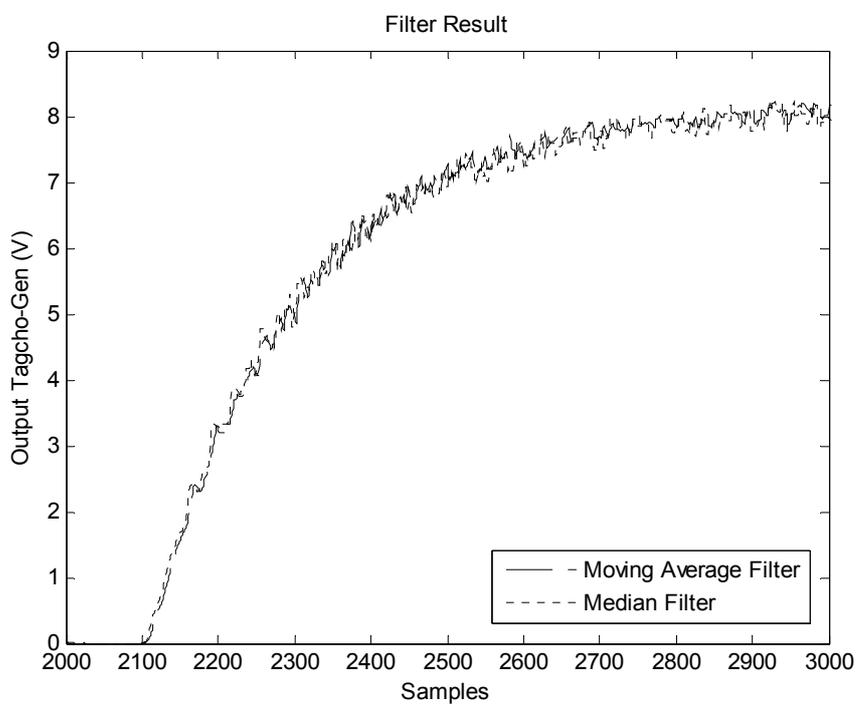
ทำการ Plot กราฟเพื่อทำการเปรียบเทียบโดยใช้คำสั่ง โปรแกรม MATLAB ดังนี้

```
>> figure (1);
>> hold;
>> title ('Filter Result')
>> plot (signal1_raw_step, 'black')
>> plot (mov_s1, '--black ')
>> plot (med_s1, ':black')
>> xlabel ('Samples')
>> ylabel ('Output Tagcho-Gen (V)')
>> legend ('Raw Data', 'Moving Average Filter', 'Median Filter')
>> axis ([2100 2400 0 9])
>> hold;
```

ผลที่ได้แสดงดังภาพที่ 44 (1) จากภาพดังกล่าวจะพบว่า Moving Average Filter และ Median Filter สามารถลด Spike Noise และ Noise ทั่วๆไปได้ แต่ Moving Average Filter จะทำให้เสมือนมี Delay ของสัญญาณเกิดขึ้นดังแสดงจากกราฟของสัญญาณที่ผ่าน Moving Average Filter แล้วเกิดขึ้นแยกออกไปทางขวาของกราฟของสัญญาณข้อมูลดิบ แต่ Median Filter ไม่ทำให้เกิด Delay ของสัญญาณ จะเห็นว่าในกรณีนี้ Median Filter ทำงานได้ดีกว่าจึงจะใช้ข้อมูลที่ได้จาก Median Filter นี้ทำการวิเคราะห์ในขั้นต่อไป



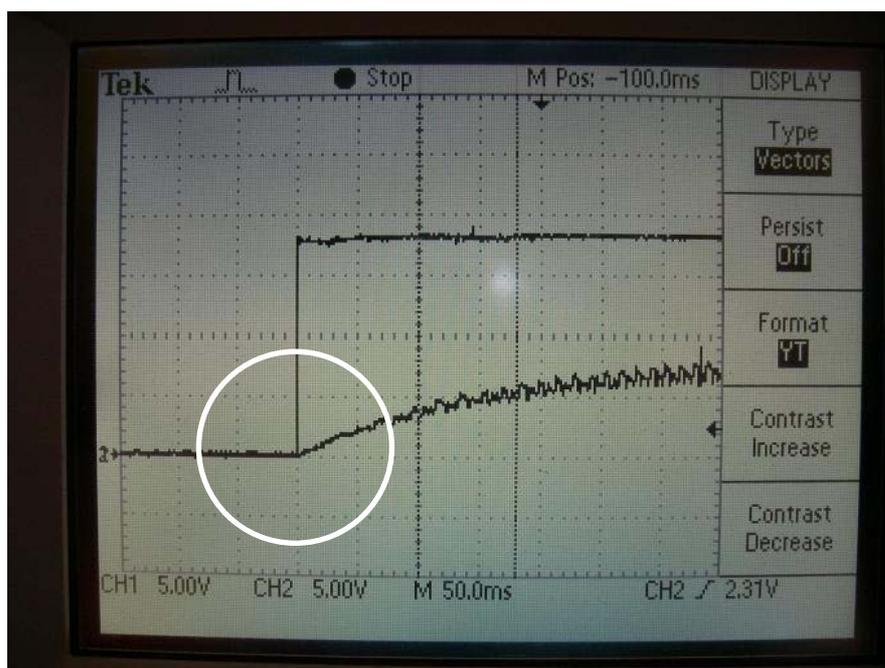
(1) แสดงในช่วงข้อมูล 2100-2400 Sample



(2) แสดงในช่วงข้อมูล 2000-3000 Sample

ภาพที่ 44 แสดงผลการกรองโดยใช้ Moving Average และ Median Filter ขนาด Window = 10

ในการทดลองใส่สัญญาณอินพุตเป็น Step ให้กับระบบที่ผ่านมานั้นเราไม่ได้เก็บค่าของสัญญาณอินพุตที่เป็น Step เข้ามาด้วยเนื่องจากแรงดัน Step ขนาด 18V มีค่าสูงเกินกว่าที่การ์ด PCI-9112 จะรับได้ ดังนั้นจากภาพที่ 43 เราจะได้เพียงรูปร่างของสัญญาณเอาต์พุตแต่ไม่สามารถวัดค่า Delay ของระบบได้ดังนั้นเราจึงต้องทำการวัดสัญญาณอินพุตกับเอาต์พุตโดยใช้ Oscilloscope อีกครั้งหนึ่งแสดงผลตามภาพที่ 45



ภาพที่ 45 แสดงสัญญาณอินพุตเป็น Step ขนาด 18V และผลตอบสนองของระบบจาก Oscilloscope

จากภาพที่ 43 และภาพที่ 45 จะเห็นได้ว่าระบบเป็นกำลังหนึ่งที่ไม่มี Delay ดังนั้นสมการระบบกำลังหนึ่งได้ดังนี้

$$G(s) = \frac{Ka}{s+a}$$

$G(s)$  ถูกกระตุ้นด้วยสัญญาณอินพุตที่เป็น Step  $R(s) = r/s$  โดย  $r = 18V$  ดังนั้นจะได้ผลตอบสนอง  $C(s)$  แสดงได้ตามสมการดังต่อไปนี้

$$C(s) = R(s)G(s) = \frac{rKa}{s(s+a)} = \frac{rK}{s} - \frac{rK}{s+a}$$

จากภาพที่ 44 (2) ได้ค่าผลตอบสนองทางเวลาที่ 63% ของค่า Steady State เท่ากับ

$$0.63 \times 8 = 5.04 \text{ V}$$

ณ ที่ค่าผลตอบสนองมีค่าเท่ากับ 5.04 V เราสามารถประมาณค่า Time Constant ได้จากกราฟ Step Response โดยดูคร่าวๆ ได้เท่ากับ 200 ms ดังนั้นจะได้ค่าของ  $a$  คือ

$$t = 200 \times 10^{-3} = \frac{1}{a}$$

$$a = 5$$

เราสามารถหาค่า  $Ka$  ได้โดยเริ่มจากการพิจารณากราฟ Step Response จะเห็นว่าค่า Force Response มีค่าที่ Steady State เท่ากับ 8 V ซึ่งเมื่อแปลงเป็นความเร็วจะได้

$$8 \times \frac{5000}{24} = 1666.67 \text{ RPM}$$

เราสามารถหา Inverse Laplace Transform  $C(s)$  ได้ผลดังนี้

$$c(t) = c_f(t) + c_n(t) = rKa - rKae^{-at}$$

ถ้าเราพิจารณาที่เวลา Steady State นั่นก็คือที่  $t \rightarrow \infty$  จะได้  $c(t \rightarrow \infty) = 1666.67$

$$c(\infty) = c_f(\infty) + c_n(\infty) = 1666.67$$

$$= rKa - rKae^{-a \cdot \infty}$$

$$rKa = 1666.67$$

$$Ka = \frac{1666.67}{r} = \frac{1666.67}{18} = 92.59$$

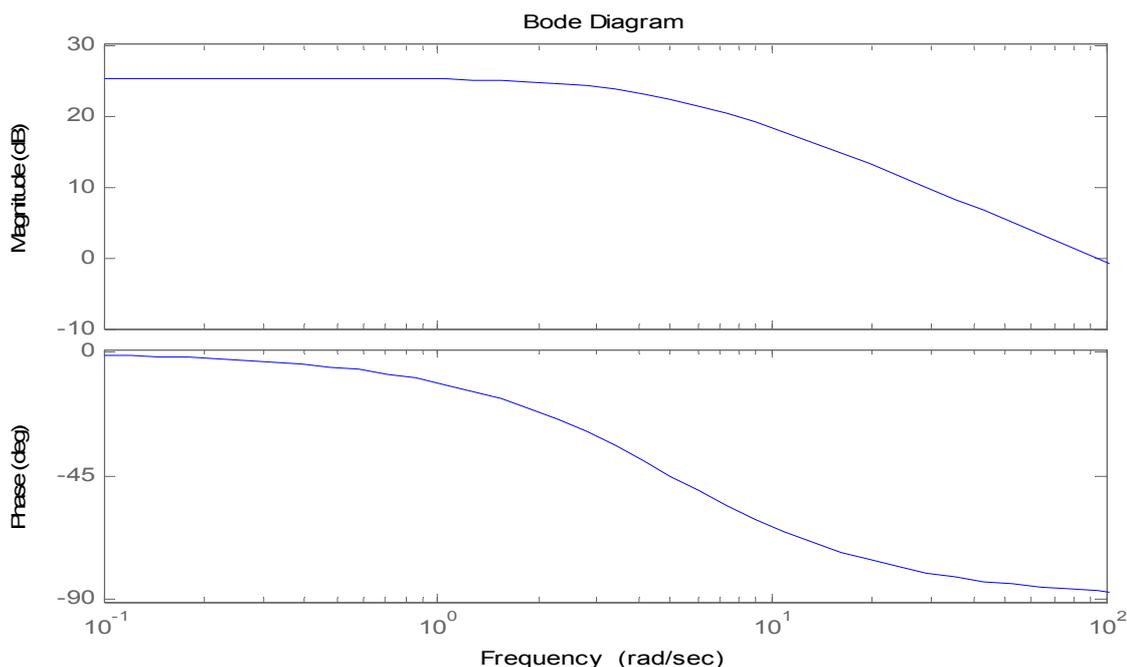
ดังนั้น Transfer Function ของระบบมอเตอร์ไฟฟ้ากระแสตรงคือ

$$G(s) = \frac{Ka}{s+a} = \frac{92.59}{s+5} \text{ RPM/V}$$

ทำการ Plot Bode ได้ผลดังภาพที่ 46 ทำได้โดยใช้คำสั่ง MATLAB ดังนี้

```
>> gc = tf ([92.59], [1 5]);
```

```
>> bode (gc)
```

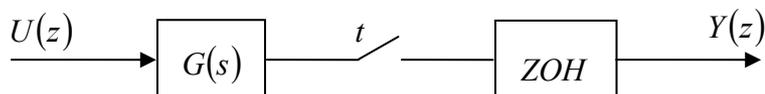


ภาพที่ 46 แสดง Bode Diagram ของ  $G(s)$

จากรูปกราฟ Bode จะเห็นว่าระบบสามารถตอบสนองได้ดีได้ที่ความถี่ประมาณไม่เกิน 10 rad/s หรือประมาณ 1.591 Hz = 0.629 s จะได้ 20 เท่าของความถี่เท่ากับ 31.82 Hz = 0.0314 s

หลังจากที่เราได้ Transfer Function ของระบบมอเตอร์ไฟฟ้ากระแสตรงซึ่งตอนนี้ยังอยู่ในรูปแบบที่เป็น Continuous ในการระบุเอกลักษณ์ของระบบนั้น เราจะต้องทำการเก็บข้อมูลที่ได้จากการทดลองโดยใช้การแปลงจากสัญญาณ Analog เป็น Digital นั่นหมายความว่า การ Sampling ของการ์ด A/D กับระบบที่เป็น Continuous ที่เรากำลังจะระบุเอกลักษณ์จะทำให้ระบบเสมือนถูกแปลงมาเป็นระบบแบบ Discrete โดยทั่วไปแล้วการ Sampling ของการ์ด A/D เป็นแบบ Zero-Order Hold ฉะนั้นการเปลี่ยนระบบจาก Continuous มาเป็น Discrete Transfer Function ของระบบก็จะต้องคูณ

ด้วย Transfer Function ของ Zero-Order Hold อีกทีหนึ่งแสดงดังภาพที่ 47 และสามารถเขียนเป็นความสัมพันธ์ได้ตามสมการ (150)



ภาพที่ 47 แสดงการ Sampling สัญญาณจาก Analog เป็น Digital

$$G(z) = \frac{Y(z)}{U(z)} = (1 - z^{-1}) Z \left\{ \frac{G(s)}{s} \right\} \quad (150)$$

การแปลง Transfer Function จาก Continuous มาเป็น Discrete โดยใช้การ Sampling แบบ Zero-Order Hold และมีคาบของการ Sampling เท่ากับ 30 ms ทำได้โดยใช้คำสั่ง MATLAB ดังนี้

```
>> gd = c2d(gc, 0.03, 'zoh')
```

ผลที่ได้คือ Transfer Function แบบ Discrete แสดงดังนี้

$$G(z) = \frac{2.579}{z - 0.8607} \quad (151)$$

เพื่อที่จะดู Order ของ  $G(z)$  เราจึงจัดรูป  $G(z)$  ให้อยู่ในรูปแบบที่เข้าใจง่าย ได้ผลดังนี้

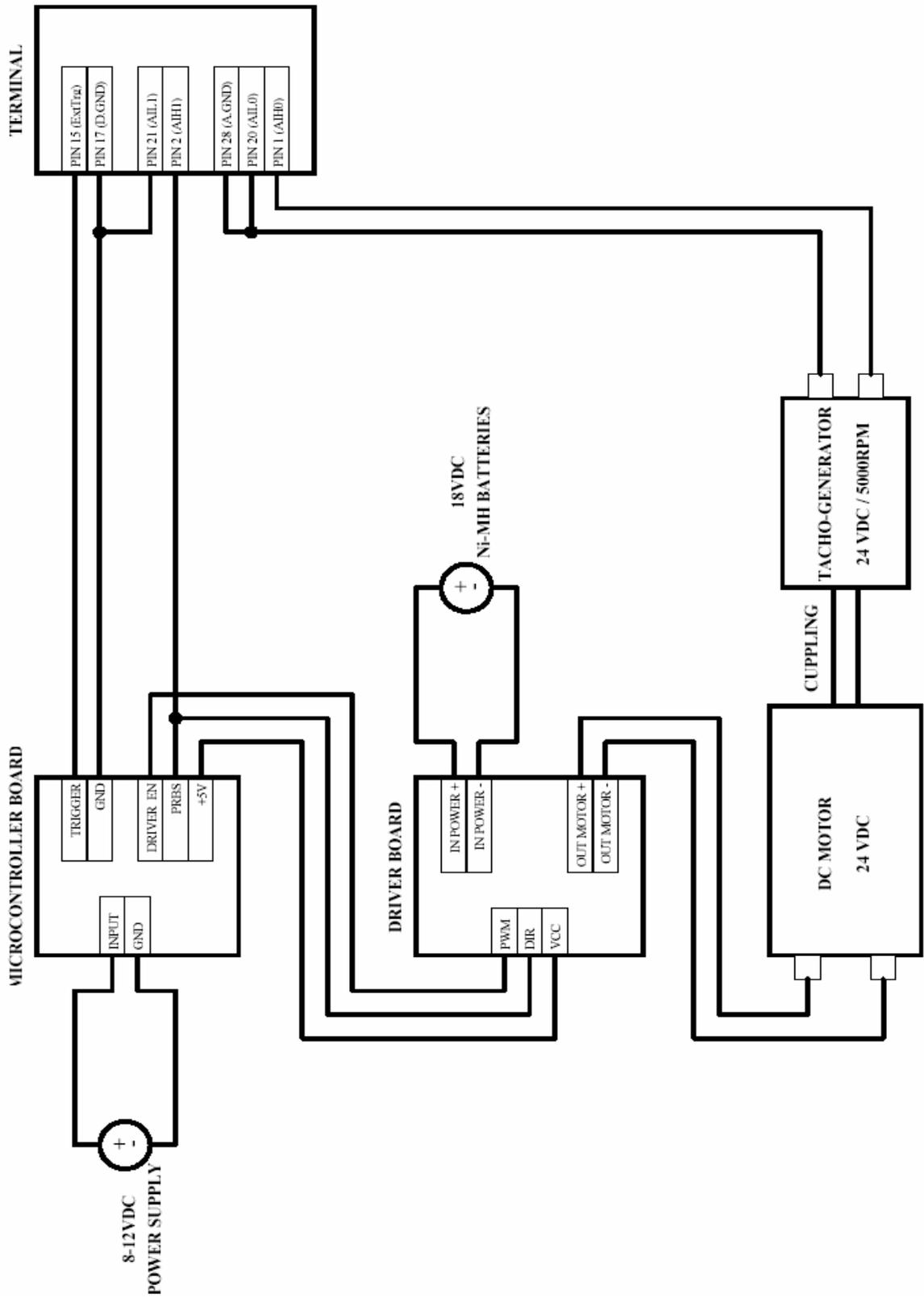
$$G(z) = \frac{Y(z)}{U(z)} = \frac{2.579z^{-1}}{1 - 0.8607z^{-1}} \quad (152)$$

จากสมการ (152) จะเห็นว่าระบบของเรามี Order ของ  $y$  เท่ากับ 1, Order ของ  $u$  เท่ากับ 1 และ Order ของ Delay เท่ากับ 1 ค่าของ Order ที่ได้นี้จะถูกนำไปใช้ในการพิจารณาเลือกค่า Order  $n_a$ ,  $n_b$  และ  $n_k$  ของแต่ละ Model Structure เพื่อใช้สำหรับการเปรียบเทียบความถูกต้องระหว่าง Model Structure แต่ละชนิด ซึ่งเราก็คาดหมายว่า Order  $n_a$ ,  $n_b$  และ  $n_k$  ที่มีค่าเท่ากับ 1 น่าจะให้ผลการระบุเอกลักษณ์ของระบบได้ดีในระดับที่น่าพอใจ

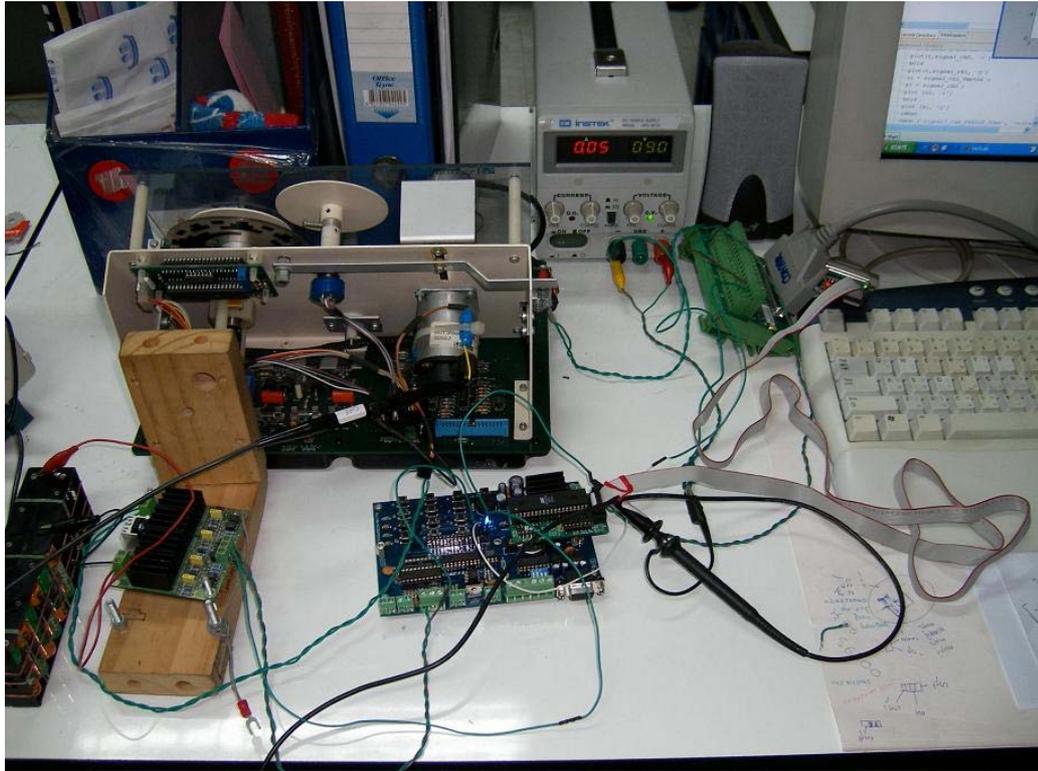
### 3. การทดลองเพื่อเก็บข้อมูลไปใช้ในการระบุเอกลักษณ์ของระบบโดยใช้สัญญาณอินพุตเป็น PRBS

เราได้ออกแบบชุดทดลองระบุเอกลักษณ์ของระบบโดยใช้สัญญาณอินพุตเป็น PRBS ตามหลักการในภาพที่ 40 โดยใช้อุปกรณ์การทดลองที่ได้กล่าวถึงไปแล้ว ซึ่งจะได้แบบไฟของวงจรแสดงตามภาพที่ 48

เราใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 ของบริษัท ATMEL เบอร์ AT82S52 ซึ่งจะทำหน้าที่สร้างสัญญาณตามที่เราโปรแกรมไว้ส่งให้กับ Driver Board เพื่อปรับเปลี่ยนสัญญาณให้พอเพียงกับการที่จับมอเตอร์ จากนั้นสัญญาณที่ถูกปรับเปลี่ยนจะถูกป้อนให้กับมอเตอร์ซึ่งจะทำให้เกิดการหมุนที่ความเร็วต่างกันแปรผันตามกับสัญญาณที่ป้อนให้กับตัวมัน สิ่งที่เราสนใจคือความเร็วรอบของมอเตอร์ที่เกิดขึ้น เราจะไม่ทำการวัดความเร็วรอบของมอเตอร์โดยตรงแต่จะใช้อุปกรณ์ที่เรียกว่า Tacho-Generator เป็นตัวแปลงความเร็วรอบของมอเตอร์มาเป็นความต่างศักย์แบบที่มีความสัมพันธ์กันเป็นเชิงเส้น ดังนั้นสัญญาณความต่างศักย์นี้เปรียบเสมือนสัญญาณเอาต์พุตของระบบมอเตอร์ ซึ่งจะถูกเก็บโดยใช้การ์ด PCI-9112 เนื่องจากเราต้องใช้ค่าอินพุตในการประมวลผลด้วย เราจึงต้องใช้การ์ด PCI-9112 วัดค่าสัญญาณอินพุตด้วย จากนั้นสัญญาณอินพุตและเอาต์พุตจะถูกส่งเข้าสู่คอมพิวเตอร์เพื่อประมวลผล ในการทดลองจริงอุปกรณ์ที่ใช้จะถูกต่อเข้าด้วยกันแสดงได้ตามภาพที่ 49



ภาพที่ 48 แบบวงจรของการทดลองระบบเอกลักษณ์ของระบบ โดยใช้สัญญาณอินพุตเป็น PRBS



ภาพที่ 49 แสดงการต่ออุปกรณ์การทดลอง

เราจะเริ่มต้นที่การสร้างสัญญาณอินพุตเพื่อทดสอบระบบโดยใช้ไมโครคอนโทรลเลอร์ โดยได้เลือกใช้สัญญาณอินพุตจากหลักการที่ว่า สัญญาณอินพุตที่เหมาะสมกับการกระตุ้นระบบจะต้องประกอบไปด้วยความถี่ต่างๆความถี่ที่ต้องการทดสอบระบบ ซึ่งสัญญาณอินพุตดังกล่าวก็คือเป็นชนิด PRBS เนื่องจากมีพฤติกรรมคล้ายกับสัญญาณรบกวนสีขาวซึ่งสามารถพิสูจน์ได้ สัญญาณ PRBS สามารถสร้างได้จากสมการ

$$\begin{aligned} u(t) &= \text{rem}(A(q)u(t), 2) \\ &= \text{rem}(a_1u(t-1) + \dots + a_nu(t-n), 2) \end{aligned}$$

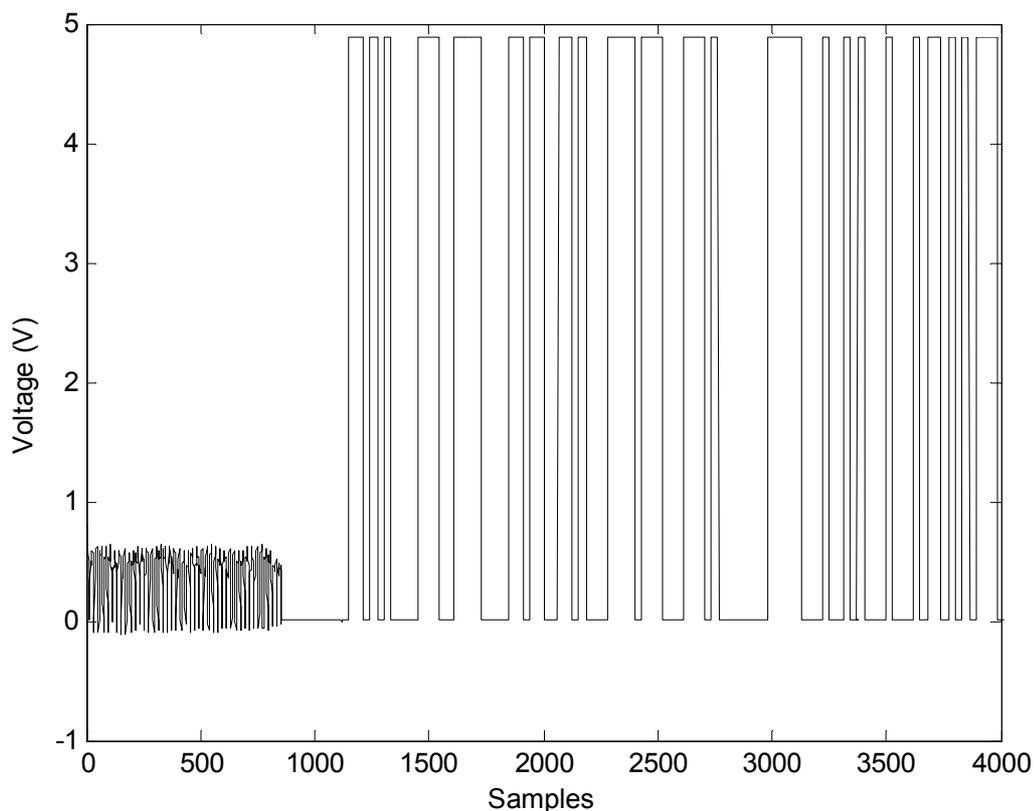
โดยมีค่าเวกเตอร์ค่าในอดีตคือ  $[u(t-1) \ u(t-2) \ u(t-3) \ \dots \ u(t-n)]$

ในตอนเริ่มต้นเวกเตอร์ค่าในอดีตนี้เราต้องสมมติค่าเริ่มต้นขึ้นเองซึ่งต้องไม่ใช่ค่าเริ่มต้นที่เป็น 0 ทำการเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์ สร้างสัญญาณ PRBS ที่มี  $n = 8$  โดยวิธีการเขียนทำได้โดยให้มองสมการของการสร้างสัญญาณ PRBS ให้อยู่ในรูปแบบของภาพที่ 25

และจะใช้ค่าระหว่างสัญญาณ PRBS ที่สร้างขึ้นแต่ละค่าคือ 0.0314 s หรือประมาณ 30 ms นั้น  
 หมายความว่าระบบเราจะถูกกระตุ้นด้วยความถี่ตั้งแต่ 0 Hz จนถึงประมาณ  $31.82/2 = 15.91$  Hz

จากการทดลองหาค่าความถี่ตอบสนองของมอเตอร์โดยใช้สัญญาณอินพุตเป็น Step พบว่าความถี่ที่ระบบตอบสนองได้มีค่าเพียง 1.591 Hz ดังนั้นค่าองค์ประกอบของความถี่ภายในสัญญาณ PRBS ที่เราสร้างขึ้นมีค่าสูงสุดสูงกว่าค่าความถี่ตอบสนองของมอเตอร์ประมาณ 10 เท่า ก็แสดงว่าความถี่ของสัญญาณ PRBS ก็น่าจะเพียงพอต่อการทดสอบระบบมอเตอร์

เราสร้างสัญญาณ PRBS ดังกล่าวโดยใช้โปรแกรมภาษา C โปรแกรมดังกล่าวชื่อ Gen\_PRBS.c แสดงในภาคผนวก ค โปรแกรม Gen\_PRBS.c จะใช้ค่าเริ่มต้น (Initial Condition) ในการสร้างสัญญาณ PRBS เท่ากับ  $0x3f = 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1$  จะได้สัญญาณ PRBS ตามภาพที่ 50 โดยจากภาพดังกล่าวสัญญาณ PRBS ที่เก็บได้จะเริ่มต้นที่ Sample 1455 เป็นต้นไป



ภาพที่ 50 แสดงข้อมูลที่ได้จากการเก็บค่าสัญญาณ PRBS ที่สร้างจากไมโครคอนโทรลเลอร์โดยใช้โปรแกรม Syn\_Muti\_V\_Edit.m (Sampling Rate 1 KHz)

นอกจากโปรแกรม Gen\_PRBS.c จะสร้างสัญญาณ PRBS แล้ว โปรแกรมยังสร้างสัญญาณอีก 2 ชนิด คือ สัญญาณ TRIGGER ใช้เป็นสัญญาณ External Trigger ให้กับการ์ด PCI-9112 เพื่อที่จะเป็นตัวบอกให้การ์ด PCI-9112 อ่านค่าสัญญาณ PRBS กับสัญญาณเอาต์พุตของระบบเข้าไปพร้อมๆกัน สัญญาณ DRIVER\_EN ใช้เป็นสัญญาณคำสั่งให้กับบอร์ด Driver เพื่อให้บอร์ดเปลี่ยนระดับแรงดันของสัญญาณและปล่อยแรงดันให้กับระบบ ความสัมพันธ์ของสัญญาณต่างแสดงเป็น Timing Diagram ได้ตามภาพผนวกที่ 4

สัญญาณ PRBS จากไมโครคอนโทรลเลอร์จะไปเข้าสู่การปรับเปลี่ยนระดับแรงดันของสัญญาณที่ Driver Board ซึ่งจะเปลี่ยนสัญญาณจากระดับ 0 V. เป็น -18 V. และจากระดับ +5 V. เป็น +18 V. แสดงดังภาพที่ 51 แรงดัน  $\pm 18$  V จะถูกส่งออกไปเป็นสัญญาณอินพุตให้กับมอเตอร์ ทำให้มอเตอร์เกิดการเคลื่อนที่แบบหมุนเปลี่ยนแปลงไปอย่างต่อเนื่องตามลักษณะของสัญญาณอินพุตที่ได้รับ

ระหว่างที่มอเตอร์เกิดการเคลื่อนที่ขึ้น การเคลื่อนที่จะถูกถ่ายทอดไปยังแกนโรเตอร์ของ Tacho-Generator ทำให้เกิดความต่างศักย์ขึ้นที่ขั้วของ Tacho-Generator ทำให้เราสามารถวัดความเร็วการหมุนของมอเตอร์ได้โดยการวัดความต่างศักย์ที่ขั้วของ Tacho-Generator สำหรับในการทดลองนี้เราจะต้องทำการเก็บค่าความต่างศักย์ที่เกิดขึ้นที่ขั้วของ Tacho-Generator พร้อมกับกับเก็บค่าของสัญญาณ PRBS ที่สร้างจากไมโครคอนโทรลเลอร์ ณ เวลาเดียวกันให้ได้และการเก็บค่าดังกล่าวจะเกิดขึ้นก่อนเมื่อมีการส่งค่าของสัญญาณ PRBS ออกมาใหม่ เราสามารถทำให้เหตุการณ์ดังกล่าวเกิดขึ้นได้โดยต่อเชื่อมวงจรให้เป็นไปตามภาพที่ 48 และใช้โปรแกรม Gen\_PRBS.c ในการสร้างสัญญาณร่วมกับการเก็บข้อมูลโดยใช้โปรแกรม MATLAB ชื่อ Syn\_Muti\_V\_Edit.m โดยจะต้องทำการกำหนดค่าของตัวแปรต่างๆใหม่ดังนี้

- TrigSource = 2                   คือ กำหนดให้ใช้ Clock ภายนอกเป็นตัวสร้างความถี่ในการ Sampling ในที่นี้คือสัญญาณ TRIGGER จะใช้เป็นสัญญาณ External Trigger เพื่อเป็นตัวบอกให้การ์ด PCI-9112 ทำการเก็บค่าเมื่อใดบ้าง

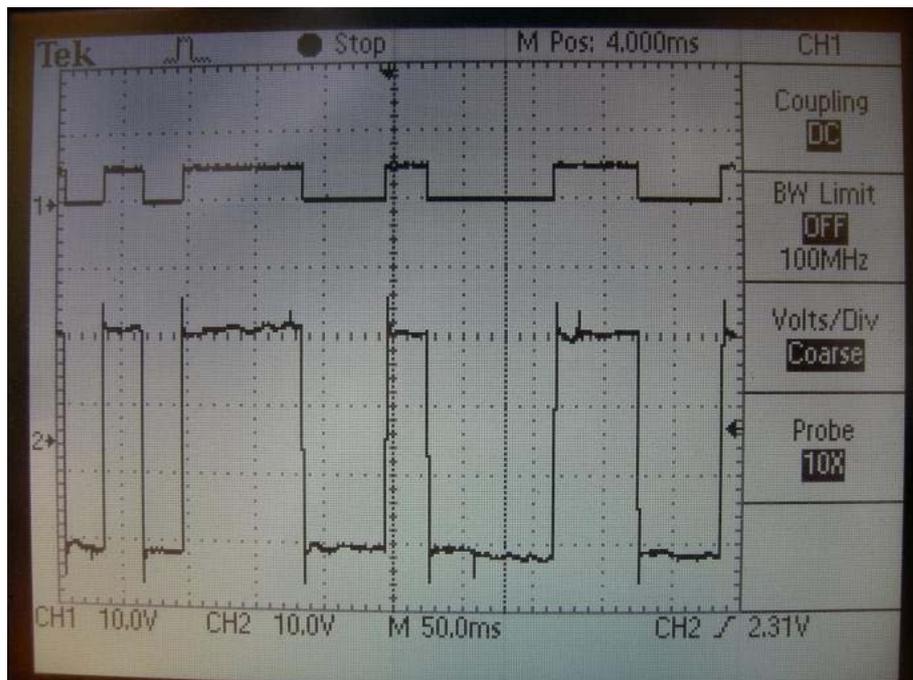
- ai\_buf = 5126                   คือ กำหนดค่าของพื้นที่ๆใช้เก็บข้อมูล

- data\_size = 2048               คือ กำหนดค่าของพื้นที่ๆใช้เก็บข้อมูล

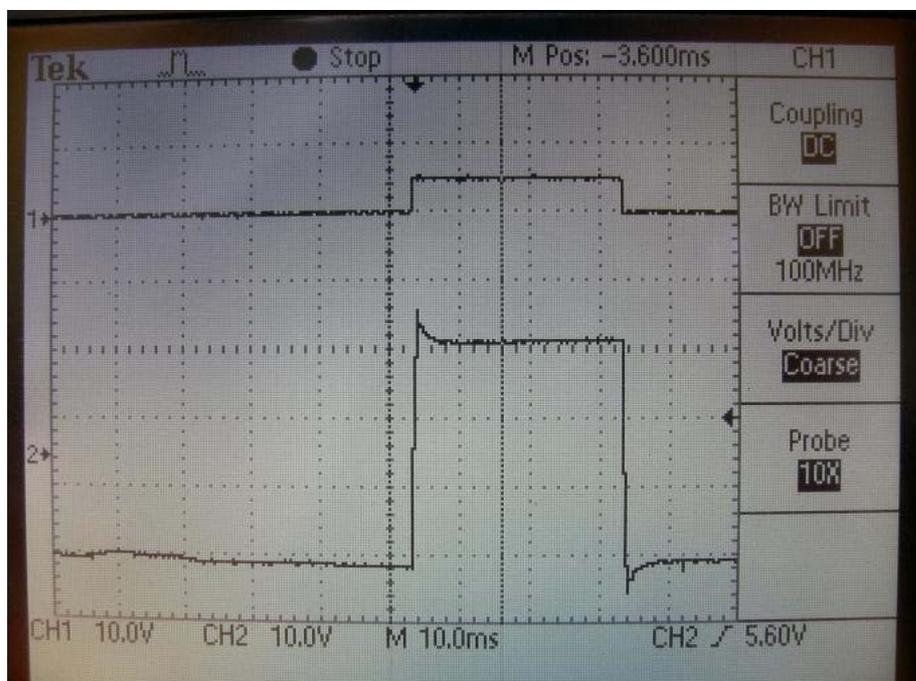
จากการกำหนดค่าของตัวแปรจะพบว่าเราจะไม่จำเป็นต้องกำหนดค่าของตัวแปร `sample_rate` เพราะในกรณีที่กำหนดให้ `TrigSource = 2` ค่าของตัวแปร `sample_rate` ก็จะไม่มีการทำงานของการ์ด PCI-9112 ตามที่ได้กล่าวมาแล้วสัญญาณ PRBS สร้างขึ้นโดยใช้  $n = 8$  ทำให้ความยาวคาบของสัญญาณ PRBS คือ  $M = 2^8 - 1 = 255$  ฉะนั้นถ้าเรากำหนดให้ตัวแปร `data_size` มีขนาดเท่ากับ 2048 เพื่อใช้สำหรับเก็บข้อมูลของ 2 ช่องสัญญาณนั้นก็หมายความว่าเราจะเก็บข้อมูลการทดลองของแต่ละช่องสัญญาณได้ 1024 ค่า นั่นก็คือเราเก็บข้อมูลการทดลองได้ประมาณ 4 คาบของสัญญาณ PRBS

หลังจากที่กำหนดค่าตัวแปรต่างๆเรียบร้อยแล้ว ให้ทำการ Run โปรแกรม ผลที่ได้คือ การ์ด PCI-9112 จะทำการเก็บข้อมูลสัญญาณที่ช่องสัญญาณที่ 0 และช่องสัญญาณที่ 1 โดยจะเก็บก็ต่อเมื่อได้รับสัญญาณ TRIGGER จากไมโครคอนโทรลเลอร์ กระบวนการเก็บข้อมูล 1024 ค่า ระยะเวลาระหว่างข้อมูลแต่ละค่าคือ 30 ms ดังนั้นก็ควรจะใช้เวลาเก็บประมาณ  $1024 \times 30 \text{ ms} = 30.72 \text{ s}$  จากแต่เนื่องจากการ์ด PCI-9112 มีกระบวนการประมวลผลอย่างอื่นเข้ามาด้วยจึงทำให้ต้องใช้เวลารวมในกระบวนการเก็บข้อมูลทั้งหมดเกิน 30.72 s แต่ก็จะไม่เกิน 1 นาที จากรูปแบบไฟงจรภาพที่ 48 ข้อมูลของช่องสัญญาณที่ 0 คือผลตอบสนองของระบบหรือเอาต์พุตของระบบซึ่งจะถูกเก็บไว้ในตัวแปร `signal_ch0` และข้อมูลของช่องสัญญาณที่ 1 คือสัญญาณ PRBS จากไมโครคอนโทรลเลอร์ซึ่งจะถูกเก็บไว้ในตัวแปร `signal_ch1`

ถ้าเราพิจารณาจากรูปแบบไฟงจรภาพที่ 48 แล้วจะเห็นว่าข้อมูลสัญญาณอินพุตในตัวแปร `signal_ch1` คือสัญญาณ PRBS ที่มีระดับแรงดัน 0V และ 5V ซึ่งยังไม่ใช่แรงดันที่เข้าไปเป็นอินพุตให้กับระบบมอเตอร์ ขณะนี้เรากำลังต้องการข้อมูลที่เป็นสัญญาณอินพุตให้กับระบบมอเตอร์เราจึงควรจะวัดสัญญาณจากขั้วของมอเตอร์โดยตรงแต่ทำไม่ได้เพราะแรงดัน  $\pm 18 \text{ V}$  มีค่าสูงเกินกว่าที่การ์ด PCI-9112 จะรับได้ การที่เราวัดสัญญาณตามแบบไฟงจรภาพที่ 48 เราจะได้ข้อมูลที่ไว้สำหรับระบุเอกลักษณ์ของระบบมอเตอร์ร่วมกับระบบ Driver ด้วย แต่จากภาพที่ 51 เราพบว่า Driver มีพฤติกรรมเป็นเพียงตัวขยายสัญญาณเท่านั้นและไม่ทำให้เกิด Delay ของสัญญาณขึ้นทำให้สัญญาณที่ออกจาก Driver ยังมีคุณสมบัติเช่นเดียวกับสัญญาณ PRBS ดังนั้นข้อมูลสัญญาณอินพุตที่ได้จากการวัดโดยใช้แบบไฟงจรภาพที่ 48 เมื่อนำมาปรับแรงดันอินพุตโดยใช้โปรแกรมจากระดับ 0 V เป็น -18 V และจากระดับ +5 V เป็น +18 V ก็เปรียบเสมือนกับข้อมูลสัญญาณอินพุตที่ได้จากการวัดจากขั้วของมอเตอร์โดยตรง



(ก) 50 ms/DIV



(ข) 50 ms/DIV

ภาพที่ 51 แสดงสัญญาณอินพุตและสัญญาณเอาต์พุตของ Driver ที่ช่องสัญญาณ 1 และ 2 ตามลำดับ

เราจะทำการปรับเปลี่ยนระดับแรงดันโดยใช้โปรแกรม MATLAB ชื่อ Extracter.m แสดงในภาคผนวก ง ผลที่ได้จากการ Run โปรแกรม Extracter.m คือ โปรแกรมดังกล่าวจะทำการเปลี่ยนค่าของข้อมูลระดับสัญญาณในตัวแปร signal\_ch1 จากระดับ 0 V เป็น -18 V และจากระดับ +5 V เป็น +18 V แล้วเก็บไว้ในตัวแปรชื่อ signal\_ch1\_Vmotor

ขณะนี้เราได้ผลตอบสนองของระบบโดยใช้สัญญาณอินพุตเป็น PRBS อยู่ที่ตัวแปร signal\_ch0 และสัญญาณอินพุตเป็น PRBS อยู่ที่ตัวแปร signal\_ch1\_Vmotor เรียบร้อยแล้วและเราสามารถเก็บข้อมูลการทดลองแบบถาวรได้โดยเก็บข้อมูลดังกล่าวลงใน Hardisk และเพื่อความเข้าใจถึงที่มาของข้อมูล ในการเก็บข้อมูลเราจึงตั้งชื่อข้อมูลดังกล่าวเป็นชื่อเดียวกันกับการทดลอง ซึ่งสามารถทำได้โดยใช้คำสั่งโปรแกรม MATLAB ดังนี้

```
>> save ('Extract_signal2_raw_PRBSn8_30ms', 'signal_ch0', 'signal_ch1', 'signal_ch1_Vmotor');
```

ตัวแปร 3 ตัวคือ signal\_ch0, signal\_ch1 และ signal\_ch1\_Vmotor จะถูกเก็บในชื่อของ Extract\_signal2\_raw\_PRBSn8\_30ms.mat ในบริเวณ Current Directory ที่โปรแกรม MATLAB ใช้งานอยู่ในขณะนั้น

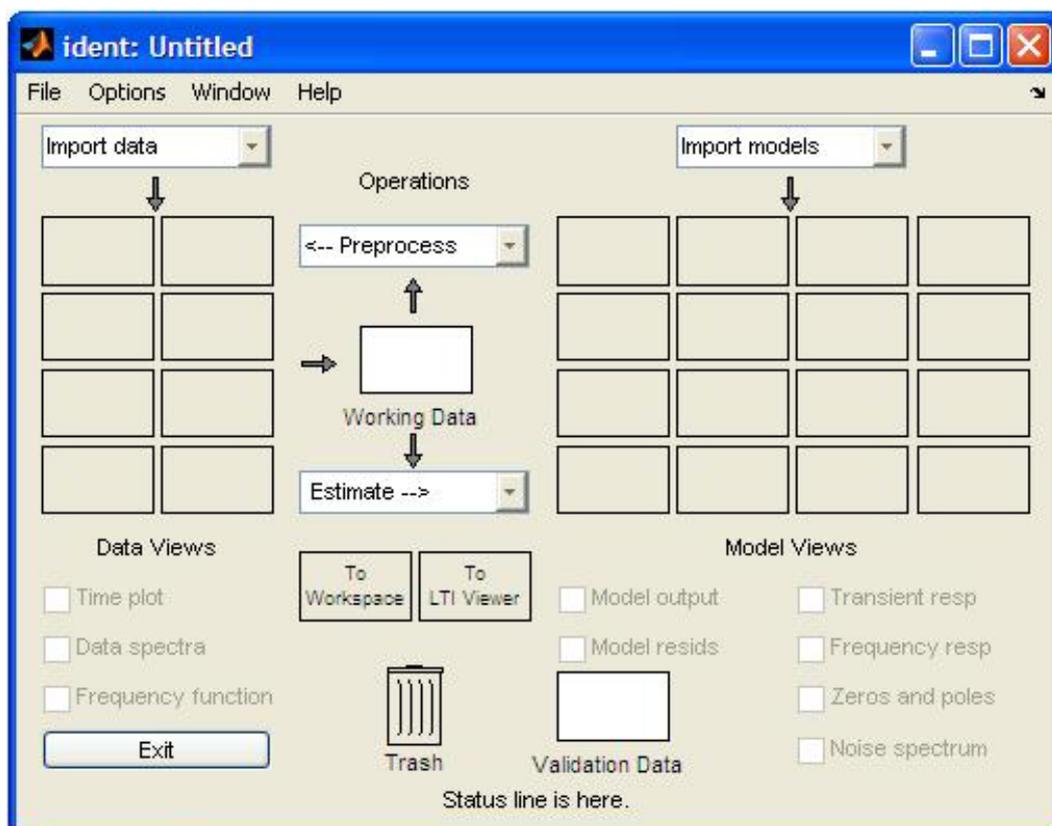
#### 4. การตรวจสอบและเตรียมข้อมูลเพื่อนำไปใช้ในการระบุเอกลักษณ์

ณ ขณะนี้เราได้ทำการทดลองเพื่อเก็บข้อมูลเรียบร้อยแล้ว ต่อไปจะเข้าสู่กระบวนการวิเคราะห์ข้อมูลดังกล่าวเพื่อนำไปใช้ในการระบุเอกลักษณ์โดยคำสั่งโปรแกรม MATLAB ดังนี้ กำหนด Current Directory ของโปรแกรม MATLAB ไปอยู่ที่เดียวกันกับที่เก็บข้อมูล Extract\_signal2\_raw\_PRBSn8\_30ms.mat จากนั้นพิมพ์คำสั่ง

```
>> load ('Extract_signal2_PRBSn8_30ms');
```

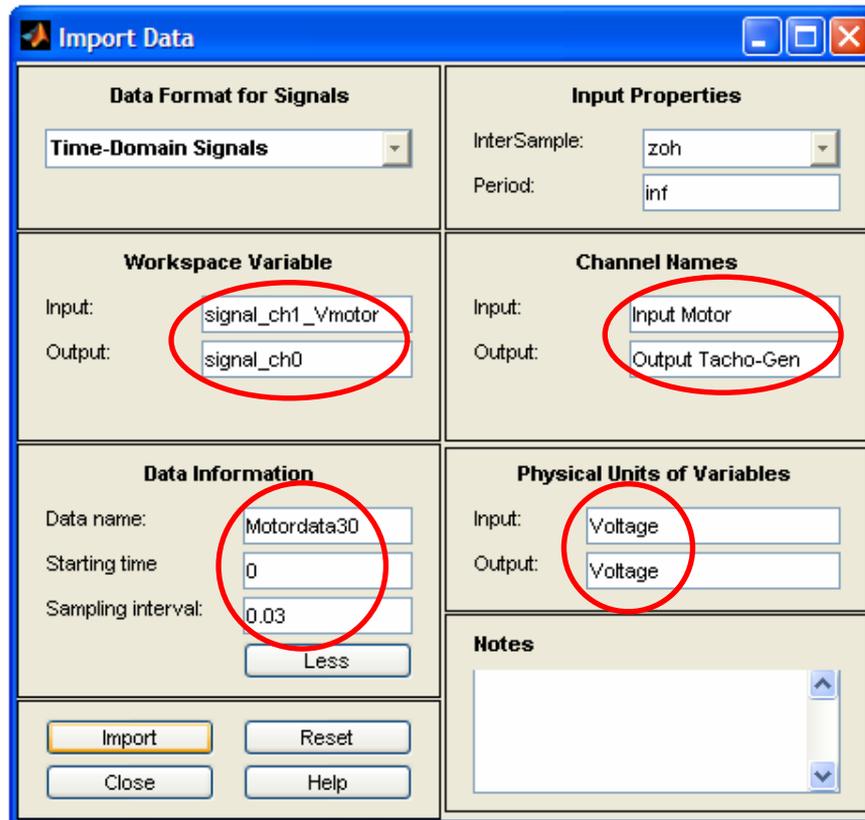
หลังการนำข้อมูลเข้ามาให้พิมพ์คำสั่ง ident ใน Command Window ผลที่ได้คือโปรแกรมจะเปิดหน้าต่าง GUI System Identification Toolbox สำหรับการระบุเอกลักษณ์ แสดงดังภาพที่ 52

```
>> ident
```



ภาพที่ 52 แสดงหน้าต่างของ ident

จากนั้นเราจะนำข้อมูลการวัดเข้าไปใน GUI System Identification Toolbox ทำได้โดยการเลือกที่ Import data -> Time domain data... ผลที่ได้คือจะ โปรแกรมจะเปิดหน้าต่างชื่อ Import Data ขึ้นมา จากนั้นให้คลิก More เพื่อให้เราสามารถใส่ค่าข้อมูลเพิ่มเติมได้ และเราจะทำการใส่ค่าข้อมูลต่างๆ ตามภาพที่ 53

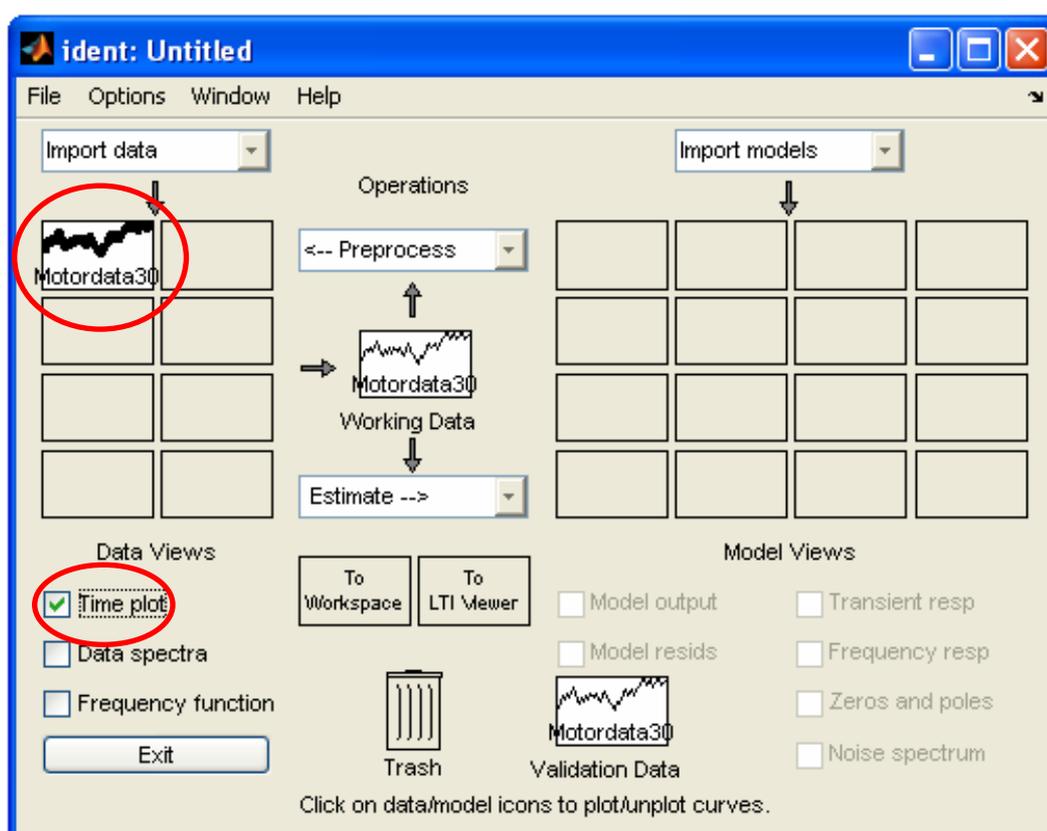


ภาพที่ 53 แสดงหน้าต่าง Import Data หลังจากกำหนดค่าต่างๆเรียบร้อยแล้ว

ทำการใส่ค่าตามภาพที่ 53 ซึ่งแต่ละค่ามีความหมายดังนี้

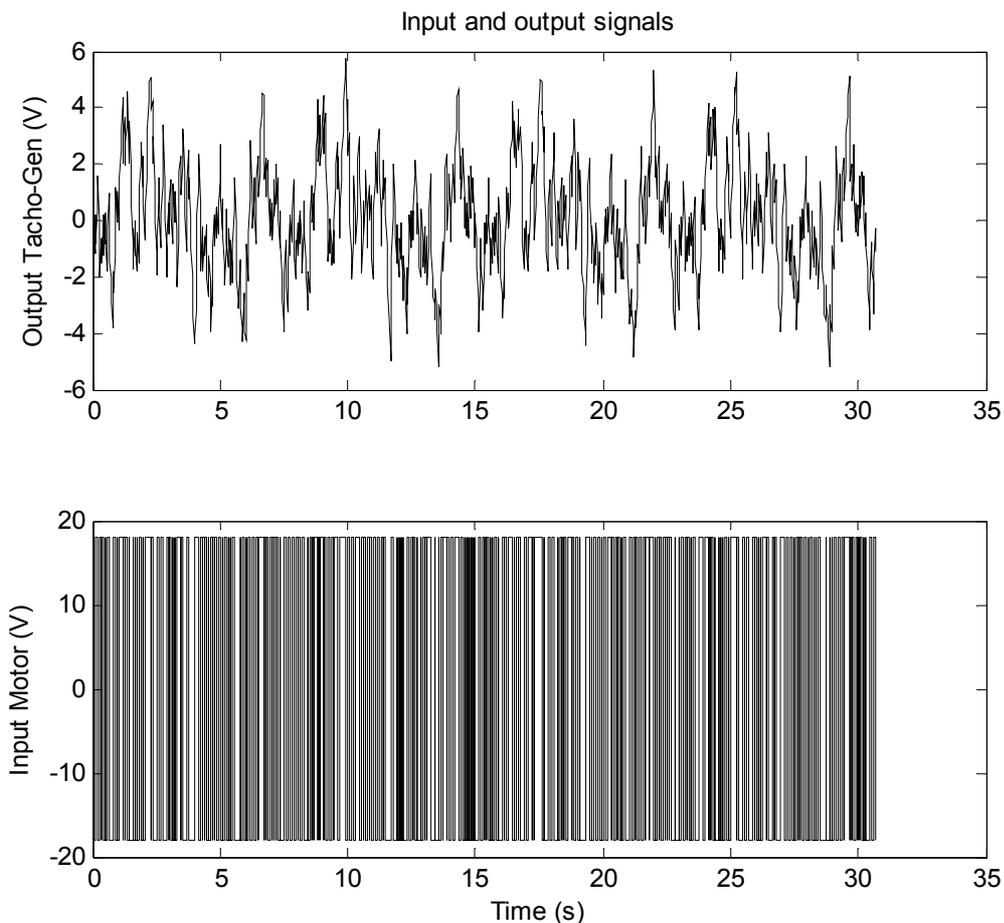
- Workspace Variable คือ ชื่อของตัวแปรที่เก็บค่าสัญญาณอินพุต-เอาต์พุตของระบบ
- Data name คือ ตั้งชื่อของข้อมูลอินพุตและเอาต์พุต
- Starting time คือ กำหนดค่าเวลาเริ่มต้นที่เก็บข้อมูลอินพุตและเอาต์พุต
- Sampling interval คือ กำหนดระยะเวลาระหว่างข้อมูลอินพุตและเอาต์พุตแต่ละค่า
- Channel Names คือ ชื่อของสัญญาณอินพุต-เอาต์พุตของระบบ
- Physical Units of Variables คือ หน่วยของสัญญาณอินพุต-เอาต์พุตของระบบ

จากนั้นเลือก Import เพื่อเป็นการนำตัวแปร signal\_ch0 และตัวแปร signal\_ch1\_Vmotor มารวมกันเป็นชุดข้อมูล (Object) ชื่อ Motordata30 และจะถูกนำเข้าสู่หน้าต่าง ident จากนั้นทำการปิดหน้าต่าง Import Data ข้อมูลที่นำเข้าไปจะอยู่ในส่วนของ Data Views ในส่วนนี้จะแสดงข้อมูลการทดลองที่นำมาจากภายนอกและข้อมูลการทดลองที่ได้จากการ Preprocess เราสามารถทำการตรวจสอบข้อมูลที่เรานำเข้ามาก็คือผลที่ได้จากการเก็บข้อมูลสัญญาณอินพุตเป็นสัญญาณ PRBS และเอาท์พุตจาก Tacho-Generator เราสามารถทำการตรวจสอบข้อมูลที่อยู่ในส่วนของ Data Views ได้โดยการเลือก Time plot ที่หน้าต่าง ident แสดงผลดังภาพที่ 54



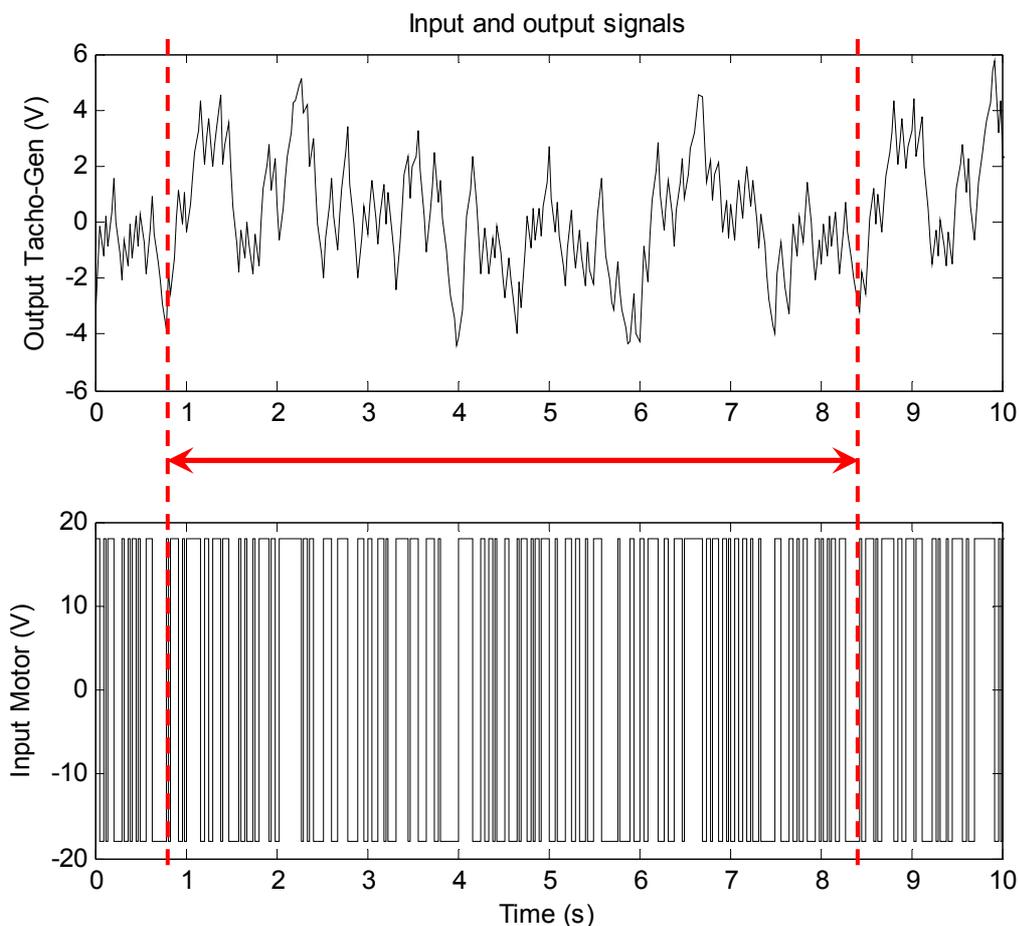
ภาพที่ 54 แสดงหน้าต่างหลังจากนำข้อมูลในส่วน Data Views เข้ามาในส่วน Working Data แล้วทำการเลือก Time plot แล้ว

หลังการเลือก Time plot โปรแกรมจะเปิดหน้าต่าง Time plot ขึ้นมาเพื่อแสดงผลแสดงข้อมูลการทดลองทั้งสัญญาณอินพุตและสัญญาณเอาท์พุตทั้งหมดที่อยู่ในส่วน Data Views แสดงกราฟในหน้าต่าง Time plot ดังภาพที่ 55



ภาพที่ 55 แสดงสัญญาณอินพุตและเอาต์พุตของระบบที่เก็บได้จากการทดลอง

เราสามารถตรวจสอบความเหมาะสมหรือความเป็นไปได้เบื้องต้นของระหว่างข้อมูลสัญญาณอินพุตและข้อมูลสัญญาณเอาต์พุตว่าถูกต้องหรือไม่ได้โดยดูจากความสัมพันธ์ระหว่างข้อมูลดังกล่าวในช่วงเวลา 1 คาบของสัญญาณ ทำได้โดยการขยายภาพเพื่อพิจารณาผลตอบสนองในช่วงของสัญญาณ PRBS จำนวน 1 คาบ คือประมาณ  $255 \times 30 \text{ ms} = 7.65 \text{ s}$  แสดงดังภาพที่ 56 ซึ่งเราจะเห็นได้จากภาพที่ 56 ว่า สัญญาณอินพุต PRBS ที่ซ้ำกันเป็นคาบก็จะทำให้เกิดผลตอบสนองที่ซ้ำกันมีลักษณะเป็นคาบเช่นกัน การตรวจสอบข้อมูลเบื้องต้นนี้แสดงให้เห็นว่าข้อมูลเหล่านี้มีความเหมาะสมในการทดลอง



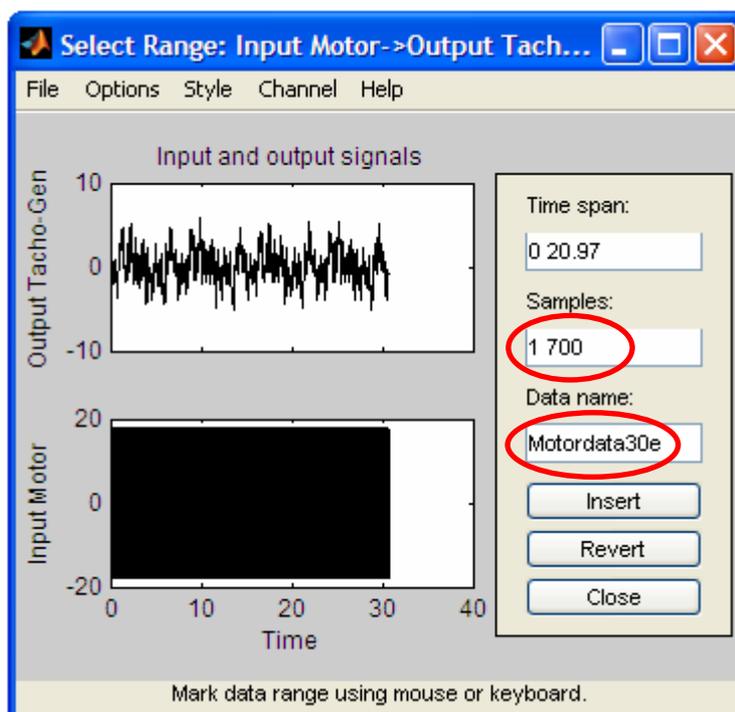
ภาพที่ 56 แสดงสัญญาณอินพุตและเอาต์พุตของระบบช่วงเวลา 0 ถึง 10 วินาที

หลังจากที่เราได้ตรวจสอบความเหมาะสมของข้อมูลชื่อ Motordata30 เบื้องต้นเรียบร้อยแล้ว ก่อนที่เราจะนำข้อมูลดังกล่าวเข้าไปใช้ทำการระบุเอกลักษณ์ เราจะทำการแบ่งข้อมูลออกเป็น 2 ส่วนก่อนคือ ส่วนของข้อมูลที่ใช้สำหรับการระบุเอกลักษณ์และส่วนของข้อมูลที่ไว้ใช้ทดสอบความถูกต้องของแบบจำลองทางคณิตศาสตร์ที่ได้จากการระบุเอกลักษณ์ สามารถทำได้โดยการนำข้อมูลที่เราต้องการจะแยกส่วน จากที่อยู่ในส่วนของ Data Views เข้ามาไว้ในส่วนของ Working Data ซึ่งทำได้โดยการ Drag and Drop จากนั้นที่หน้าต่าง ident เลือก Preprocess -> Select range... ผลที่ได้คือจะโปรแกรมจะเปิดหน้าต่างชื่อ Select Range ขึ้นมาเพื่อให้เราใส่ค่าข้อมูลต่างๆ เราจะทำการแบ่งส่วนของข้อมูลเป็น 2 ส่วนดังนี้

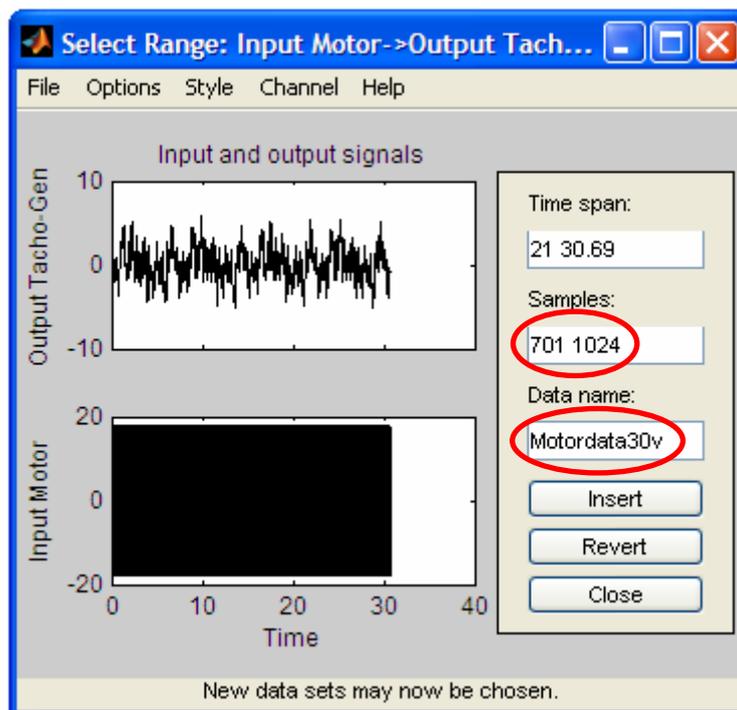
1. แบ่งข้อมูลส่วนหนึ่งมาใช้สำหรับการระบุเอกลักษณ์ เรียกข้อมูลส่วนนี้ว่า Estimation Data เราทราบมาแล้วว่าข้อมูลที่ได้จากการทดลองมีลักษณะเป็นคาบหลายคาบ ดังนั้นข้อมูลที่เราจะ

นำมาใช้ Estimate ก็ควรจะเลือกจำนวนอย่างน้อย 1 คาบของชุดข้อมูลเพื่อที่จะได้องค์ประกอบของผลตอบสนองครบถ้วน สัญญาณ PRBS ที่เราใช้ในการทดลองมีความยาวคาบเท่ากับ 255 นั่นก็คือเราควรจะเลือกจำนวนข้อมูลที่จะนำมาใช้ Estimate มากกว่าค่าดังกล่าว ในที่นี้เราจะเลือกจำนวน 700 Samples จากข้อมูลที่มีอยู่จำนวน 1024 Sample มาใช้สำหรับการ Estimate ทำได้โดยการใส่ค่าตามตามภาพที่ 57 จากนั้นเลือก Insert เพื่อเป็นการนำข้อมูลชื่อ Motordata30e เข้าไปในส่วนของ Data Views

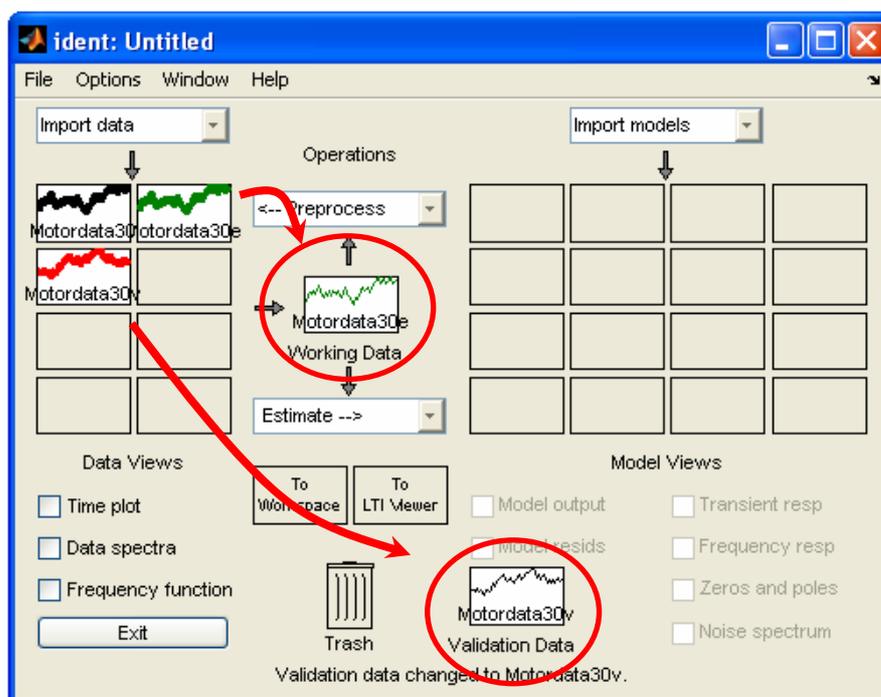
2. แบ่งข้อมูลอีกส่วนหนึ่งมาใช้เป็นข้อมูลสำหรับทดสอบแบบจำลองทางคณิตศาสตร์ เรียกข้อมูลส่วนนี้ว่า Validation Data ข้อมูลในส่วนนี้ต้องไม่ใช่ข้อมูลที่อยู่ในข้อมูลที่ใช้ในการ Estimate และควรจะเป็นข้อมูลที่มีรูปแบบหลากหลาย ถ้าเราใช้ข้อมูลส่วนนี้เป็นสัญญาณ PRBS ก็ควรจะเลือกจำนวนอย่างน้อย 1 คาบสัญญาณเพื่อที่จะสามารถทดสอบแบบจำลองทางคณิตศาสตร์ได้ทุกอย่างองค์ประกอบของความถี่ในสัญญาณ PRBS ในที่นี้เราเลือกข้อมูล 324 Samples คือตั้งแต่ข้อมูลที่ Sample 701 ถึงข้อมูลที่ Sample 1024 สามารถทำได้โดยการใส่ค่าตามตามภาพที่ 58 จากนั้นเลือก Insert เพื่อเป็นการนำข้อมูลชื่อ Motordata30v เข้าไปในส่วนของ Data Views และทำการปิดหน้าต่าง Select Range



ภาพที่ 57 แสดงหน้าต่างของ Select Range ขณะกำหนดช่วงของข้อมูลที่เราจะนำมาใช้ Estimate



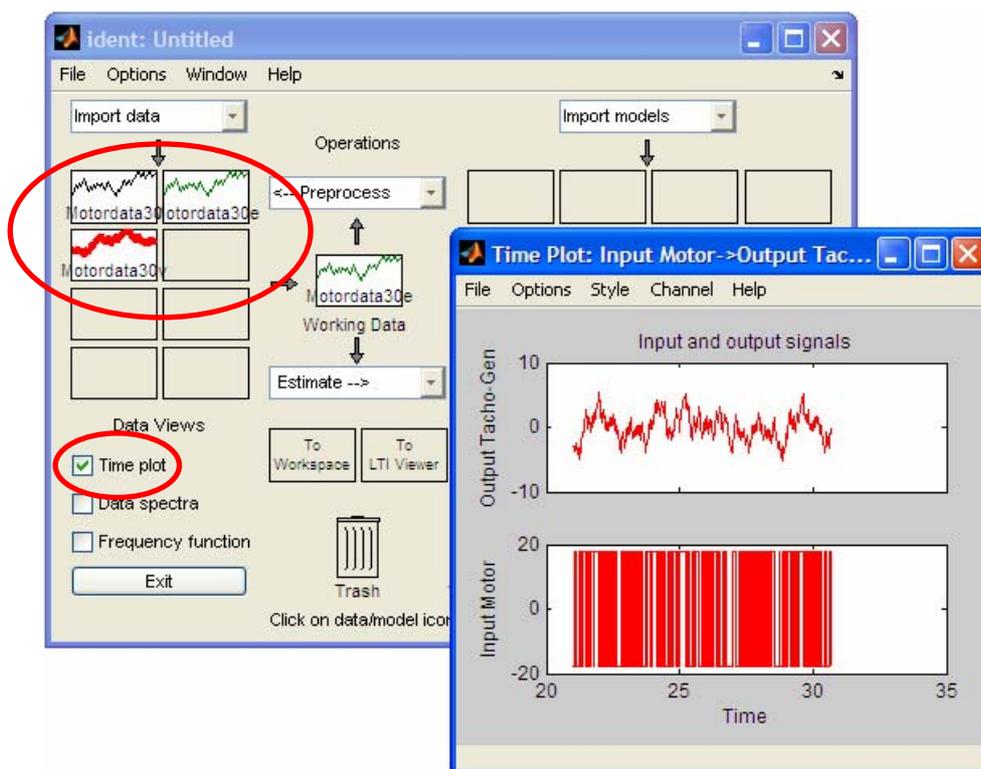
ภาพที่ 58 แสดงหน้าต่างของ Select Range ขณะกำหนดช่วงของข้อมูลที่เราจะนำมาใช้ทดสอบ ความถูกต้องของแบบจำลองทางคณิตศาสตร์



ภาพที่ 59 แสดงหน้าต่างเตรียมพร้อมข้อมูลสำหรับการระบุเอกลักษณ์

หลังจากที่เราแบ่งข้อมูลไว้สำหรับ Estimation และ Validation แล้ว เราจะต้องทำการเตรียมข้อมูลดังกล่าวไว้สำหรับการระบุเอกลักษณ์และสำหรับการทดสอบแบบจำลองทางคณิตศาสตร์ ที่ทำได้โดยนำข้อมูลในส่วน Data Views ชื่อ Motordata30e ไปไว้ในส่วน Working Data และ Motordata30v ไปไว้ในส่วน Validation Data ซึ่งทำได้โดยการ Drag and Drop ผลที่ได้แสดงในภาพที่ 59

ในหน้าต่าง ident เราสามารถตรวจสอบข้อมูลที่เราได้ทำการแบ่งส่วนไว้เรียบร้อยแล้วโดยการเลือก Time plot ตามที่ได้กล่าวมาแล้ว และเรายังสามารถกำหนดได้ด้วยว่าเราต้องการให้ข้อมูลชุดไหนบ้างที่จะนำไป Plot กราฟโดยการคลิกที่บริเวณในกรอบของข้อมูลชุดที่ต้องการในส่วน Data View ข้อมูลที่จะถูกนำไป Plot กราฟจะแสดงด้วยเส้นสีหนา ตัวอย่างการแสดงความต้องการให้ Plot กราฟเฉพาะข้อมูลชื่อ Motordata30v แสดงดังภาพที่ 60



ภาพที่ 60 แสดงการ Plot กราฟเฉพาะชุดข้อมูลชื่อ Motordata30v

สำหรับการระบุเอกลักษณ์ของระบบมอเตอร์ โดยใช้ข้อมูลที่ได้เตรียมพร้อมไว้แล้วในภาพที่ 59 จะแสดงในส่วนของผลการทดลองในหน้า 110 ต่อไป

## 5. การเก็บข้อมูลเพิ่มเติมเพื่อนำไปใช้ในตรวจสอบความถูกต้อง (Validation)

หลังจากที่เราได้เก็บข้อมูลสำหรับการใช้ในการ Estimation และ Validation ในเบื้องต้นแล้ว เราได้เก็บข้อมูลสำหรับการทำ Validation เพิ่มเติม จำนวน 2 ชุด คือ

1. ข้อมูลชื่อ Extract\_signal4\_raw\_PRBSn8\_30ms\_sampling1000Hz ข้อมูลดังกล่าวได้จากการใช้โปรแกรม Gen\_PRBS.c สร้างสัญญาณ PRBS ที่มีค่าระหว่างสัญญาณ PRBS ที่สร้างขึ้นแต่ละค่าคือ 30 ms (กำหนดค่า cnt = 117) เป็นสัญญาณอินพุตให้กับระบบมอเตอร์ แล้วใช้โปรแกรม MATLAB ชื่อ Syn\_Muti\_V\_Edit.m เก็บข้อมูลทั้งสัญญาณอินพุต-เอาต์พุต โดยใช้คาบการ Sampling เท่ากับ 1 KHz (กำหนดค่า TrigSource = 1, sample\_rate = 2000) จากนั้นเราจะทำการปรับเปลี่ยนระดับแรงดันในตัวแปรชื่อ signal\_ch1 โดยใช้โปรแกรม MATLAB ชื่อ Extarcter.m แล้วเก็บไว้ในตัวแปรชื่อ signal\_ch1\_Vmotor ขณะนี้เราได้ผลตอบสนองของระบบโดยใช้สัญญาณอินพุตเป็น PRBS อยู่ที่ตัวแปร signal\_ch0 และสัญญาณอินพุตเป็น PRBS อยู่ที่ตัวแปร signal\_ch1\_Vmotor เรียบร้อยแล้วและเราสามารถเก็บข้อมูลการทดลองแบบถาวรได้โดยเก็บข้อมูลดังกล่าวลงใน Hardisk ซึ่งสามารถทำได้โดยใช้คำสั่งโปรแกรม MATLAB ดังนี้

```
>> save ('Extract_signal4_raw_PRBSn8_30ms_sampling1000Hz', 'signal_ch0', 'signal_ch1',
        'signal_ch1_Vmotor');
```

2. ข้อมูลชื่อ Extract\_signal5\_raw\_PRBSn8\_50ms\_sampling1000Hz ข้อมูลดังกล่าวได้จากวิธีการเดียวกันกับที่ใช้สร้าง Extract\_signal4\_raw\_PRBSn8\_30ms\_sampling1000Hz ต่างกันที่ เราจะใช้โปรแกรม Gen\_PRBS.c สร้างสัญญาณ PRBS ที่มีค่าระหว่างสัญญาณ PRBS ที่สร้างขึ้นแต่ละค่าคือ 50 ms (กำหนดค่า cnt = 195) เป็นสัญญาณอินพุตให้กับระบบมอเตอร์

ในการเก็บข้อมูลเราได้ใช้คาบการ Sampling เท่ากับ 1 ms ในภายหลังเราจะได้ทราบว่า Model Structure ที่เราสร้างนั้นสร้างมาจากข้อมูลที่มีคาบในการ Sampling เท่ากับ 30 ms ดังนั้น ข้อมูลที่จะนำไปใช้ทำ Validation ก็ต้องมีคาบการ Sampling เท่ากันคือ 30 ms ด้วย ฉะนั้นข้อมูลทั้ง 2 ชุดที่เราเก็บมาเราจึงต้องทำการ Resample ก่อนถึงจะนำไปใช้ได้ นอกจากคาบการ Sampling ที่ต้องเท่ากันแล้ว ข้อมูลที่จะนำไปใช้ทำ Validation เพื่อการเปรียบเทียบความถูกต้องระหว่าง Model Structure ที่ได้ควรจะมีจำนวนที่เท่ากันด้วย ในที่นี้คือ 324 Samples

การ Resample ชุดข้อมูลโดยใช้ Function Resample ใน GUI ident จะทำให้ชุดข้อมูลถูก Resample และถูก Filter ด้วย เนื่องจากเราต้องการ Resample ชุดข้อมูลเพียงอย่างเดียวเราจึงใช้คำสั่ง ในโปรแกรม MATLAB โดยตรงแทน การ Resample ชุดข้อมูลที่มีคาบจาก 1 ms ให้เป็น 30 ms สามารถทำได้โดยกำหนด Current Directory ของโปรแกรม MATLAB ไปอยู่ที่เดียวกันกับที่เก็บ ข้อมูลชื่อ Extract\_signal4\_raw\_PRBSn8\_30ms\_sampling1000Hz.mat จากนั้นพิมพ์คำสั่ง

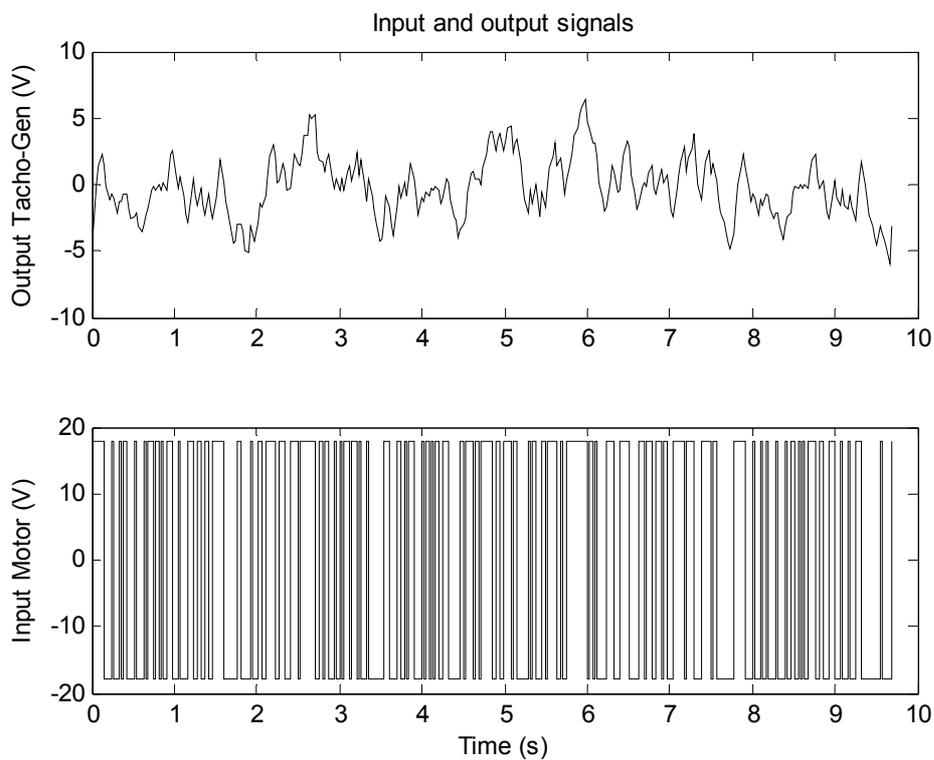
```
>> load ('Extract_signal4_raw_PRBSn8_30ms_sampling1000Hz');
>> signal_ch0_resam = resample (signal_ch0, 1, 30, 0);
>> signal_ch1_Vmotor_resam = resample (signal_ch1_Vmotor, 1, 30, 0);
```

จากนั้นเราจะเลือกข้อมูลในช่วงที่เหมาะสมมาจำนวน 324 Samples ในที่นี้เราจะเลือก ข้อมูลตั้งแต่ Sample 101 ถึง 424 เราจะกำหนดให้ข้อมูลของสัญญาณอินพุตจำนวน 324 Samples อยู่ที่ตัวแปร signal\_ch0\_resam324 และข้อมูลของสัญญาณเอาต์พุตจำนวน 324 Samples อยู่ที่ตัวแปร signal\_ch1\_Vmotor\_resam324 และจะเก็บข้อมูลอินพุต-เอาต์พุตดังกล่าวลงใน Hardisk ขั้นตอนการเลือกและเก็บข้อมูลดังกล่าวสามารถทำได้โดยใช้คำสั่งโปรแกรม MATLAB ดังนี้

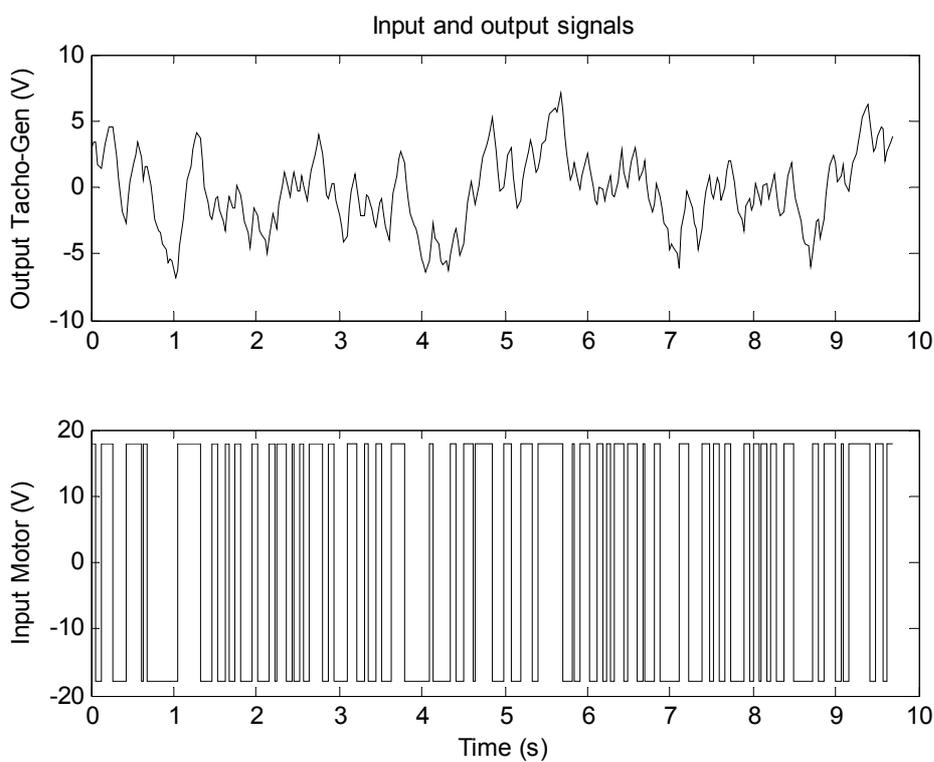
```
>> signal_ch0_resam324 = signal_ch0_resam (101:424);
>> signal_ch1_Vmotor_resam324 = signal_ch1_Vmotor_resam (101:424);
>> save ('Testdata_sig4_sampling30ms', 'signal_ch0_resam324', 'signal_ch1_Vmotor_resam324')
```

ตัวแปรอินพุต-เอาต์พุตจะถูกเก็บในชื่อ Testdata\_sig4\_sampling30ms.mat ในบริเวณ Current Directory ที่โปรแกรม MATLAB ใช้งานอยู่ในขณะนั้น เราจะทำการ Resample และเลือก ข้อมูลจำนวน 324 Samples จากข้อมูลชื่อ Extract\_signal5\_raw\_PRBSn8\_50ms\_sampling1000Hz ด้วยวิธีเดียวกันและจะ Save ในไว้ในชื่อ Testdata\_sig5\_sampling30ms.mat

ชุดข้อมูล Testdata\_sig4\_sampling30ms.mat และ Testdata\_sig5\_sampling30ms.mat จะ ถูกนำไปใช้ในการ Validation ใน GUI ident ได้โดยใช้วิธีเดียวกันกับการนำชุดข้อมูล Extract\_signal2\_raw\_PRBSn8\_30ms.mat เข้าไปใน GUI ident หลังจากการนำข้อมูลเข้าไป เรียบร้อยแล้วก็ทำการ Plot Graph แสดงตามภาพที่ 61 และภาพที่ 63 ในขณะนี้ก็เป็นอันว่าเราได้ เตรียมข้อมูลอีก 2 ชุดสำหรับการ Validation เป็นที่เรียบร้อยแล้ว



ภาพที่ 61 แสดงชุดข้อมูลชื่อ Testdata\_sig4\_sampling30ms.mat มี Sampling Interval เท่ากับ 30 ms



ภาพที่ 62 แสดงชุดข้อมูลชื่อ Testdata\_sig5\_sampling30ms.mat มี Sampling Interval เท่ากับ 30 ms

**สถานที่ทำการวิจัย**

ห้อง 2308 ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

**ระยะเวลาทำการวิจัย**

การวิจัยเริ่มตั้งแต่ เดือนตุลาคม 2549 ถึงสิ้นสุดเดือนเมษายน 2550