

LAB: System Identification

Introduction:

จากการเรียนวิชา FRA231 ที่ผ่านมาจะพบว่าในการหา Transfer function ของระบบมอเตอร์นั้น โจทย์จะกำหนดค่า Parameter ทั้งหมดไว้แล้ว แต่การจำลองระบบมอเตอร์ในความเป็นจริงนักศึกษาอาจจะไม่ทราบค่า Parameter ของ Model ของระบบมอเตอร์ที่นักศึกษาได้ออกแบบไว้ System Identification Toolbox ของ MATLAB เป็นหนึ่งในวิธีการหาค่า Parameter ของ Transfer function ของระบบมอเตอร์เพื่อให้สามารถจำลองพฤติกรรมของมอเตอร์และนำไปใช้ออกแบบระบบควบคุมต่อไปได้

วัตถุประสงค์:

- เรียนรู้และทำความเข้าใจพื้นฐานของ System Identification
- การศึกษาและประยุกต์ใช้เทคนิคต่างๆ ของ System Identification โดยใช้ MATLAB/Simulink
- เพื่อสร้างและวิเคราะห์โมเดลระบบควบคุมจากข้อมูลที่มีอยู่ ทั้งข้อมูลจำลองและข้อมูลจริง
- การทดสอบและประเมินผลโมเดลที่ได้จากขั้นตอน System Identification กับข้อมูลจริง
- การฝึกปฏิบัติและการใช้งาน MATLAB/Simulink ในการจำลองและวิเคราะห์ระบบ

แบบจำลองของมอเตอร์กระแสตรง

สมการพื้นฐานของมอเตอร์ DC มีสองส่วนหลัก: ส่วนของสมการไฟฟ้าที่ประกอบด้วย R, L และส่วนของกลที่ประกอบด้วย J, B สมการทั้ง 2 ส่วนนี้สามารถอธิบายพฤติกรรมของมอเตอร์ DC ได้ดังสมการต่อไปนี้

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + K_e \dot{\theta}$$

โดยที่ $v(t)$ คือ แรงดันไฟฟ้าที่ใช้กับมอเตอร์ (Volts)

R คือ ความต้านทานภายในของมอเตอร์ (Ohms)

$i(t)$ คือ กระแสไฟฟ้า (Amperes)

L คือ ความเหนี่ยวนำภายในของมอเตอร์ (Henrys)

K_e คือ ค่าคงที่ของ Back EMF

θ คือ ตำแหน่งเชิงมุม (rad)

สมการทางกลของมอเตอร์ DC แสดงถึงความสัมพันธ์ระหว่างแรงบิดที่ผลิตและความเร็วหมุน

$$T(t) = K_i i(t) - J\ddot{\theta} - b\dot{\theta}$$

โดยที่ $T(t)$ คือ แรงบิดของมอเตอร์ (Nm)

K_i คือ ค่าคงที่ของแรงบิดกับกระแส

b คือ ค่าความเสียดทานของมอเตอร์

J คือ โมเมนต์ความเฉื่อยของมอเตอร์

ฟังก์ชันถ่ายโอน (Transfer Function)

สมการไฟฟ้าและทางกลของ DC Motor สามารถเขียนอยู่ในรูปของ Frequency Domain ได้ดังต่อไปนี้

$$G(s) = \frac{OUTPUT}{INPUT} = \frac{K_i}{JLs^2 + (JR + Lb)s + (RB + K_e K_i)}$$

หรือ

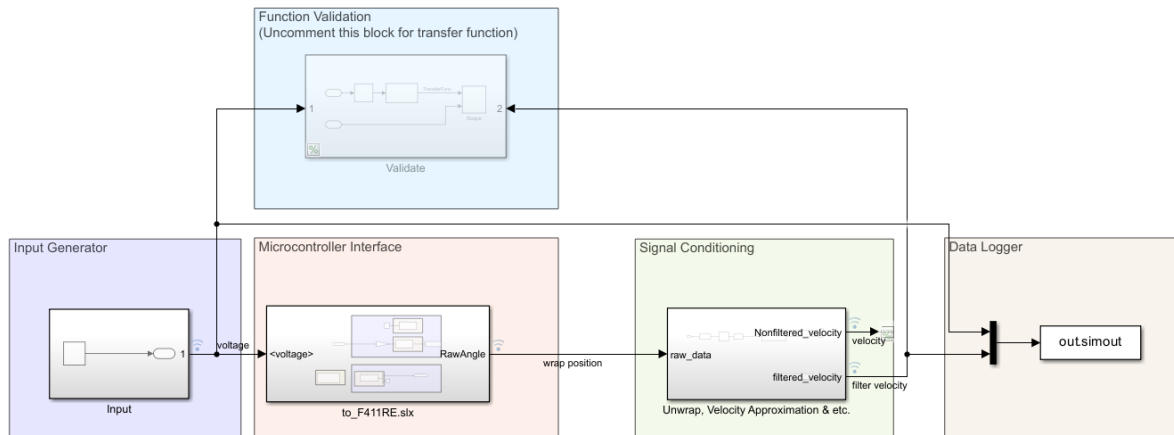
$$G(s) = \frac{OUTPUT}{INPUT} = \frac{\frac{K_i}{JL}}{s^2 + \left(\frac{R}{L} + \frac{b}{J}\right)s + \frac{RB + K_e K_i}{JL}}$$

ขั้นตอนการติดตั้ง

1. ทำการติดตั้ง DC motor และ encoder เหมือนกับ LAB2_ActuatorI
2. อัปโหลด LAB3_Firmware.bin เข้า Nucleo
3. แก้ไข COM Port ใน Block : Microcontroller Interface
4. การเก็บค่าสามารถเก็บได้จาก LAB*_interface.slx

วิธีการทดลอง

LAB*_interface.slx



****Reminder** check your com port in block "to_f411RE.slx" before Run the model!!!**

ภายในไฟล์จะประกอบไปด้วย 5 ส่วน ได้แก่

1. Input Generator สำหรับสร้างสัญญาณ input รูปแบบต่าง ๆ
2. Microcontroller Interface สำหรับเชื่อมต่อกับ Nucleo
3. Signal Conditioning สำหรับการปรับปรุงสัญญาณ
4. Data Logger ส่งค่าออกสู่ workspace
5. Function Validation สำหรับเปรียบเทียบ Transfer Function ที่หาได้จากระบบจริง

Instruction

หมายเหตุ: Instruction นี้เป็นแค่ขั้นตอนการทำกระบวนการ System Identification เบื้องต้นเท่านั้น นักศึกษาสามารถศึกษาเพิ่มเติมและทำกระบวนการอื่นๆ นอกเหนือจากที่วาง Instruction ไว้ได้

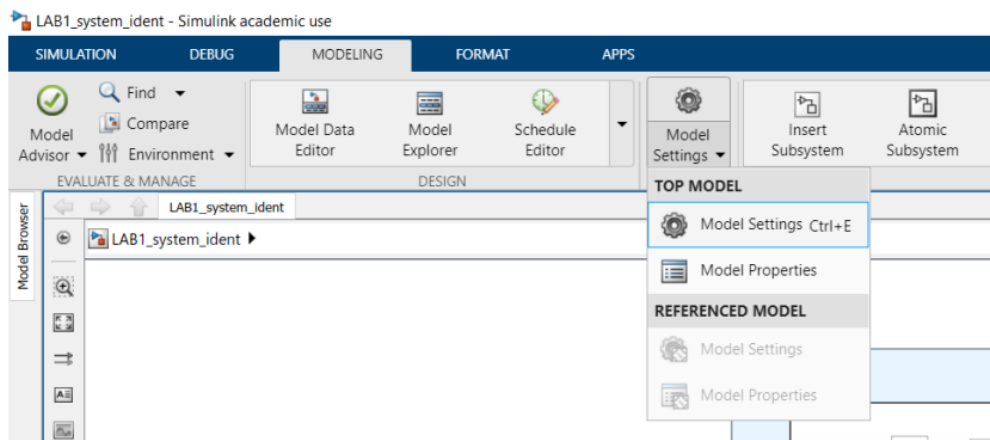
1. คำนวณหาสมการ Transfer Function ของมอเตอร์ และเนื่องระบบมีการรับค่าสัญญาณจากมอเตอร์ ภายใน MATLAB ทุก ๆ 0.01 วินาที ดังนั้น Transfer Function ต้องคำนวณเป็นระบบ Discrete Time (Domain “z”)

2. Setup บอร์ด nucleo และเชื่อมต่อบอร์ด nucleo เข้ากับคอมพิวเตอร์ในส่วน Microcontroller Interface โดยนักศึกษาสามารถศึกษาขั้นตอนเพิ่มเติมและทดลองการเชื่อมต่อได้ในเอกสาร manual_LAB1.pdf ภายในโฟลเดอร์ LAB1_firmware

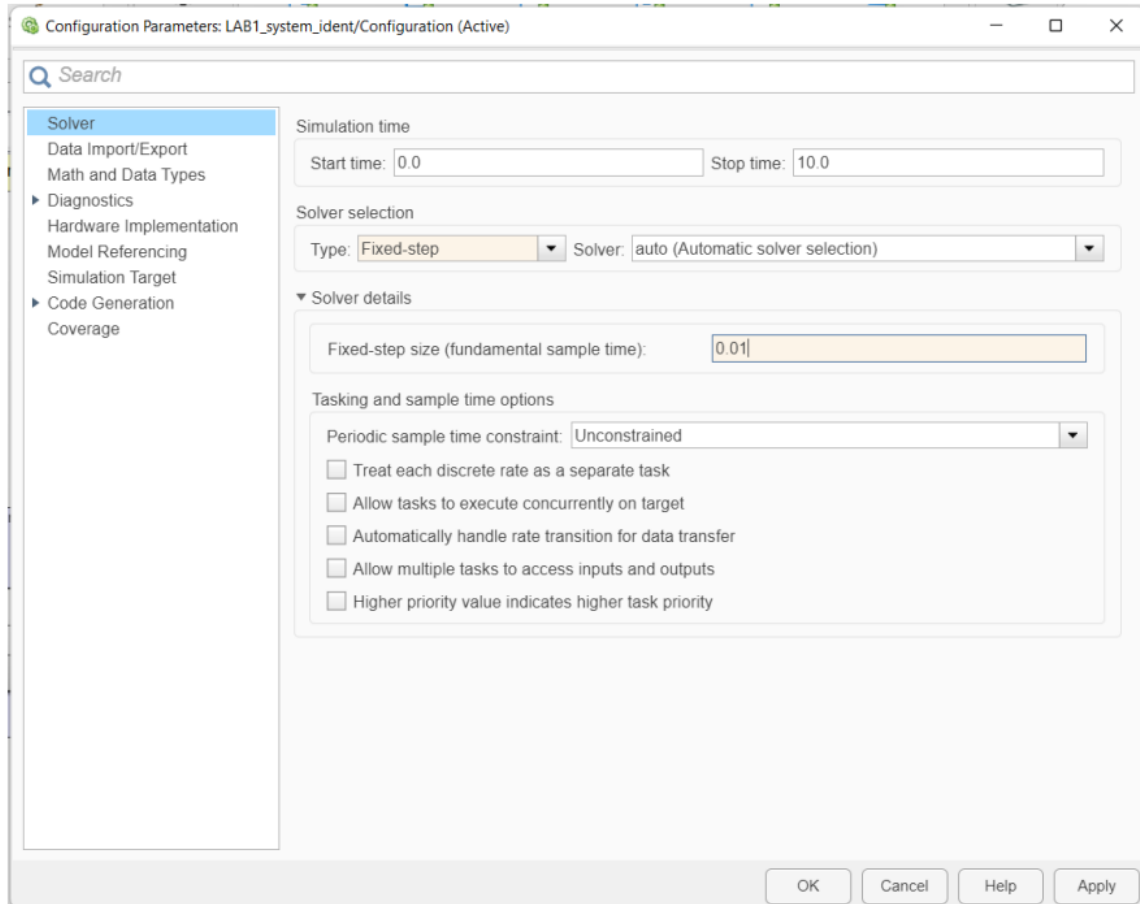
หมายเหตุ เอกสาร manual_LAB1.pdf เป็นเอกสารสำหรับเรียนรู้การใช้ Microcontroller Interface Subsystem โดยการเชื่อมต่อไม่เกี่ยวข้องกับการเชื่อมต่อโน้ตบุ๊ก LAB1_system_ident.slx

3. ทำการตั้งค่า model ใน Simulink ให้เป็น Discrete-Time โดยตั้งค่าให้ทำงานทุก ๆ 0.01 วินาที

- กด MODELING > Model Settings > Model Settings



- ตั้งค่า Type ให้เป็น Fixed-step และกำหนด Fixed-step size ให้มีค่า 0.01 จากนั้นกด Apply > OK



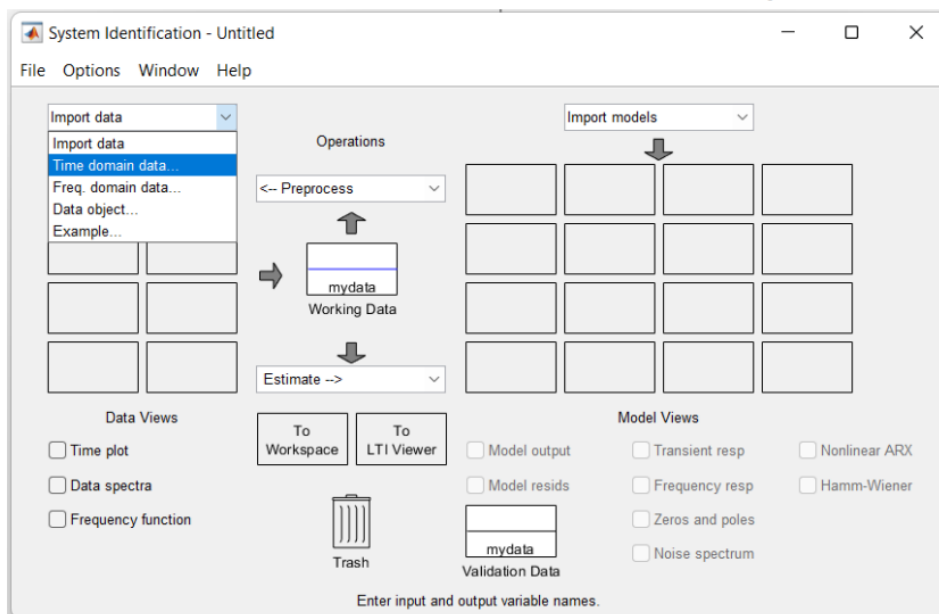
4. unwrap สัญญาณที่ได้จาก Encoder และทำการแปลงสัญญาณตำแหน่ง (output จาก unwrapping) เป็นสัญญาณความเร็วใน Unwrap & Velocity Approximate Subsystem
5. ป้อนสัญญาณ Input ใน Input Subsystem ซึ่งการที่จะให้ System Identification Toolbox สามารถประมาณค่า Parameter ให้มีความใกล้เคียงกับความเป็นจริงของมอเตอร์มากที่สุด นักศึกษา จำเป็นต้องออกแบบ ทดลองและวิเคราะห์หาสัญญาณ Input ที่มีเหมาะแก่การจำลองมากที่สุด โดยรูปแบบสัญญาณพื้นฐาน ได้แก่ sine wave, step, ramp และ chirp (สามารถใช้สัญญาณ รูปแบบอื่นที่นอกเหนือจากตัวอย่างได้)

6. กด Run เพื่อเก็บค่าสัญญาณไปยัง Workspace
7. พิมพ์คำสั่ง "ident" ในหน้าต่าง Command Window

```
Command Window
New to MATLAB? See resources for Getting Started.

>> ident
Warning: The "ident" command is obsolete and may be removed in a future release.
> In ctrlMsgUtils.warning (line 25)
In ident (line 47)
fx >>
```

8. คลิกที่หัวข้อ Import data > Time domain data (นักศึกษาสามารถ import ค่าสัญญาณได้หลายค่าได้เพื่อใช้สำหรับการทำ pre process หรือใช้สำหรับการตรวจสอบความถูกต้อง)



9. กรอกค่าของ workspace variable ลงไป จากนั้นกดปุ่ม Import

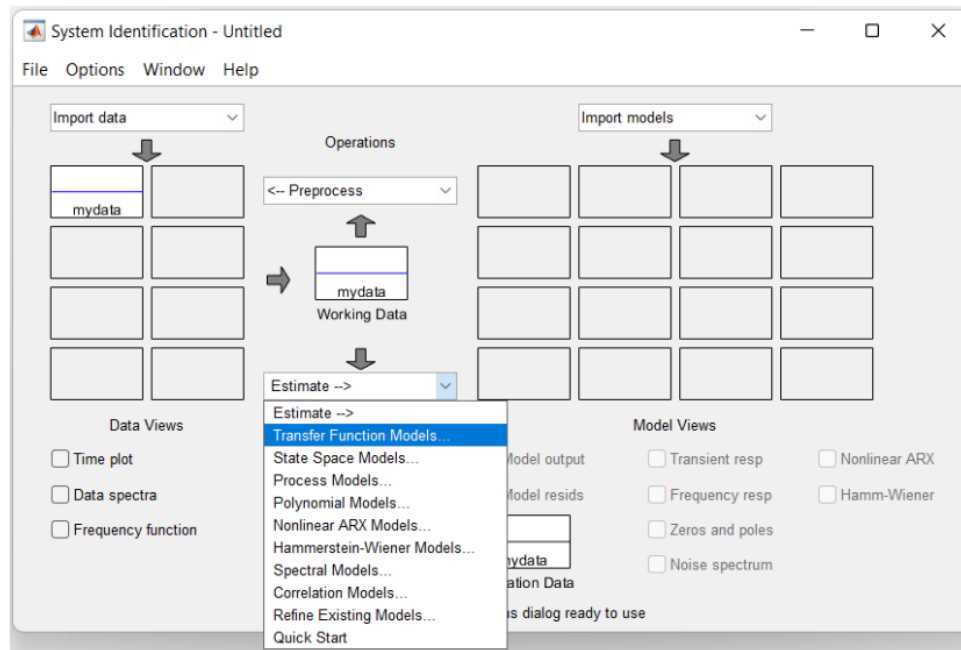
The screenshot shows the 'Import Data' dialog box with the following settings:

- Data Format for Signals:** Time Domain Signals
- Workspace Variable:**
 - Input: out.simout.Data(:,1)
 - Output: out.simout.Data(:,2)
- Data Information:**
 - Data Name: mydata
 - Start Time: 0
 - Sample time: 0.01
- Buttons:** Import, Reset, Close, Help

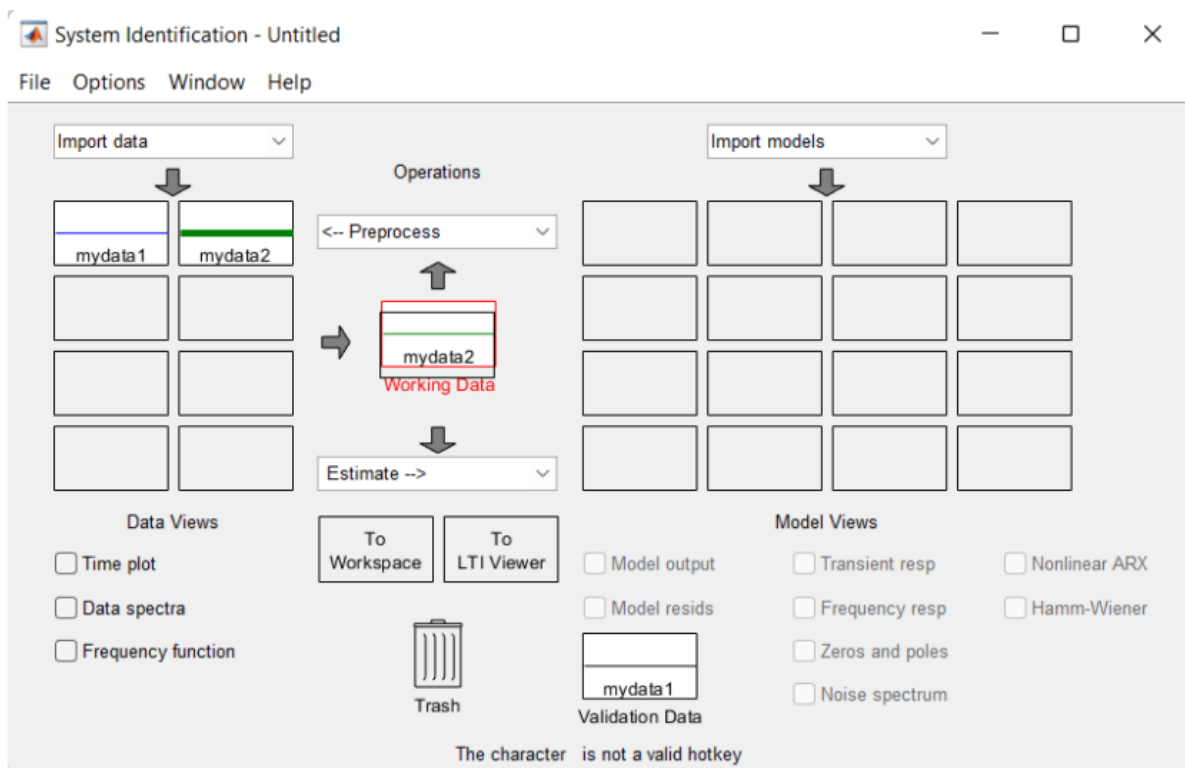
- Input คือค่าของสัญญาณที่ป้อนเข้าไปในระบบ
- Output คือค่าของสัญญาณที่ออกจากระบบ
- Data Name คือชื่อของสัญญาณ
- Start Time คือเวลาเริ่มต้นในการเก็บข้อมูล โดยกำหนดให้เริ่มที่ 0 วินาที
- Sample Time คือระยะเวลาในการเก็บข้อมูล โดยกำหนดให้เก็บข้อมูลทุกๆ 0.01 วินาที

หมายเหตุ ถ้าหากค่าสัญญาณที่เก็บค่ามานั้นมีคุณภาพไม่ดี นักศึกษาสามารถเข้าไปทำขั้นตอน Pre Process ก่อนได้

10. คลิกที่หัวข้อ Estimate > Transfer Function Model

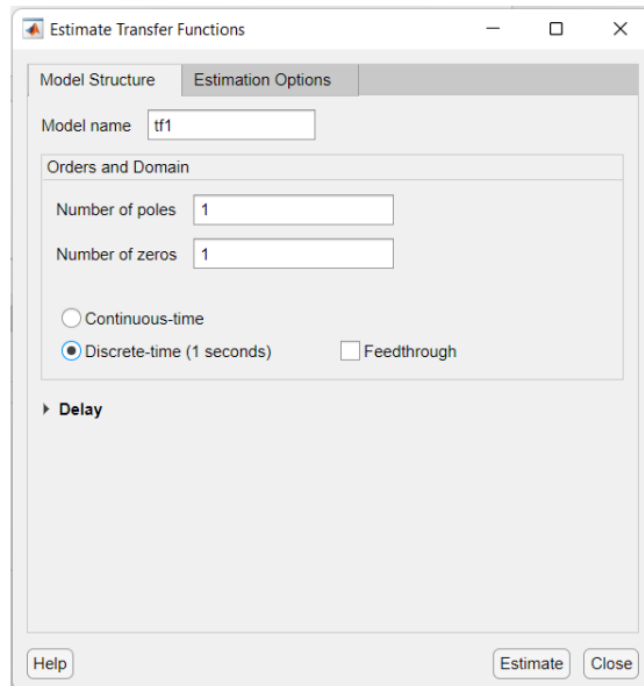


หมายเหตุ กรณีที่มี Data มากกว่า 1 ชุด นักศึกษาต้องลาก Block Data ที่ต้องการ Estimate ไปยัง Working Data



11. ประมาณค่า Transfer Function ของมอเตอร์

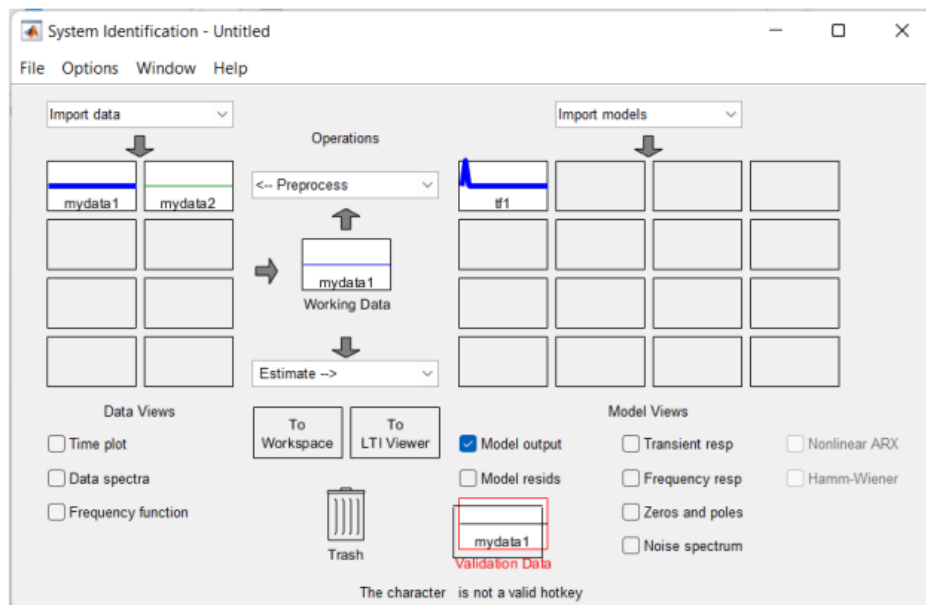
- กำหนดจำนวนของ pole และ zero ให้ถูกต้องตามสมการมอเตอร์ที่ได้จากการคำนวณ โดยต้องคำนวณเป็น Discrete-Time (Domain “z”)
- เลือก Discrete-Time (0.01 seconds)
- กดปุ่ม Estimate

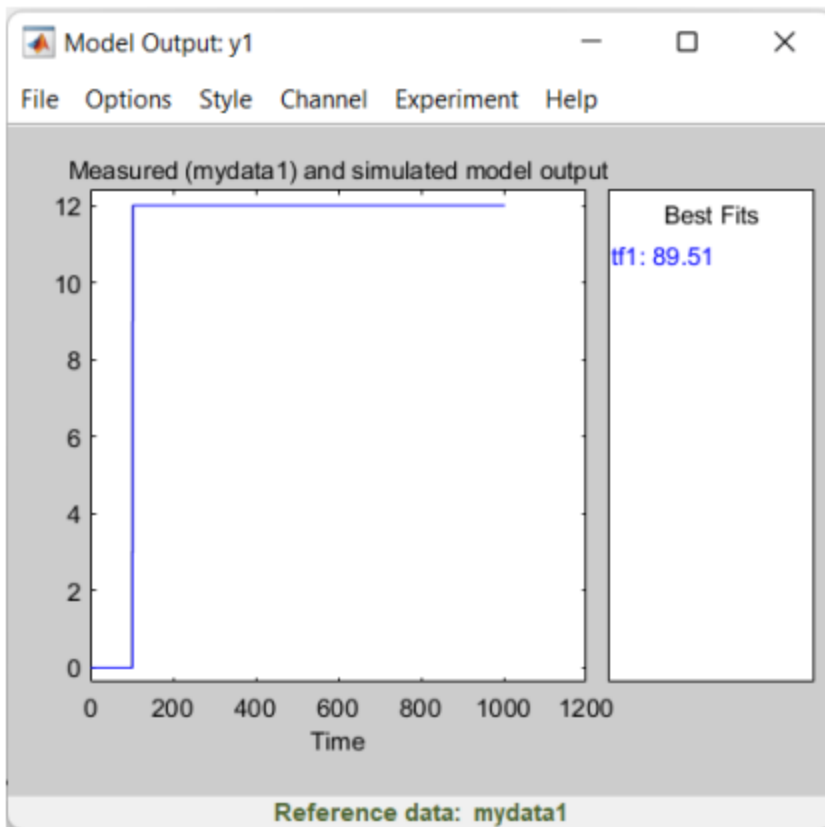


หมายเหตุ จำนวน poles กับ zeros ในภาพเป็นเพียงการยกตัวอย่างเท่านั้น ไม่ใช่จำนวนที่ถูกต้อง

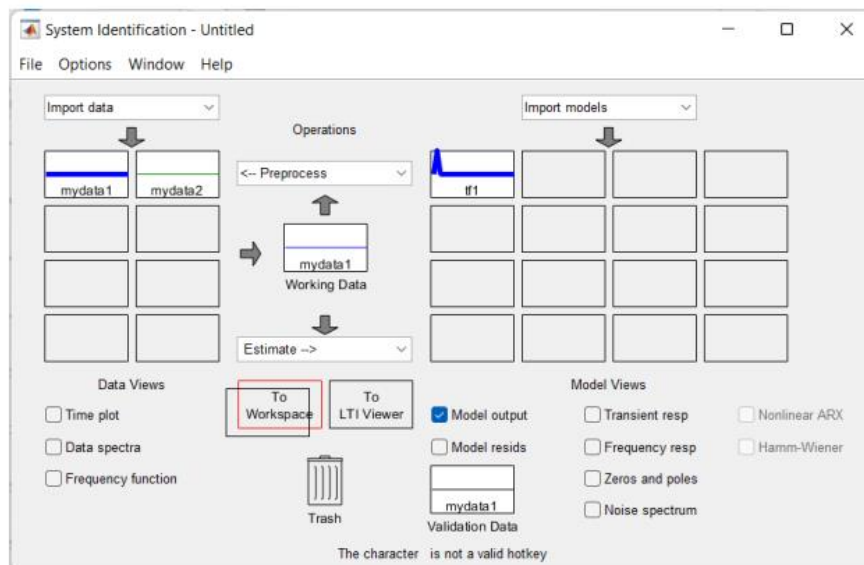
12. เมื่อได้ Transfer Function แล้วนักศึกษาสามารถตรวจสอบความถูกต้อง Transfer Function ที่ได้กับชุดข้อมูลแต่ละชุดได้ โดย

- ลาก Block Data ที่ต้องการ validate ไปยัง Validation Data
- กดปุ่ม Model Output ซึ่งจะแสดงค่าความถูกต้องระหว่าง Model ที่ถูกประมาณค่ากับค่าสัญญาณจริง

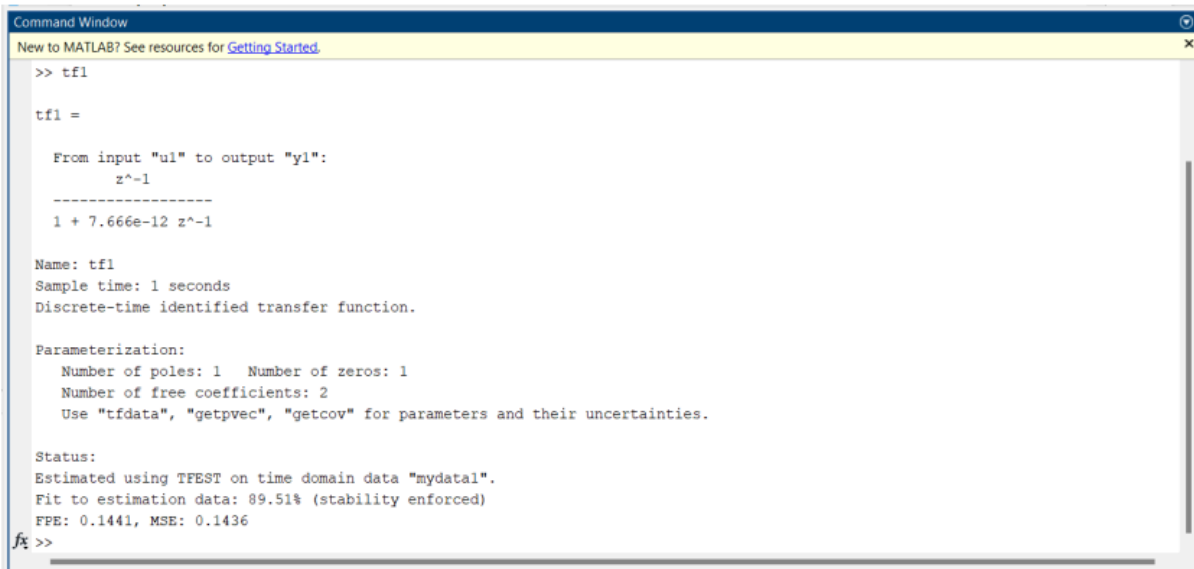




13. เมื่อนักศึกษาพอใจกับค่า Transfer Function ที่ได้แล้วนั้น จึงลาก Block tf1 ไปยังตำแหน่ง To Workspace



14. ค่า Transfer Function ที่ได้จะปรากฏที่ Workspace



```
Command Window
New to MATLAB? See resources for Getting Started

>> tfl

tfl =

    From input "u1" to output "y1":
      z^-1
    -----
    1 + 7.666e-12 z^-1

Name: tfl
Sample time: 1 seconds
Discrete-time identified transfer function.

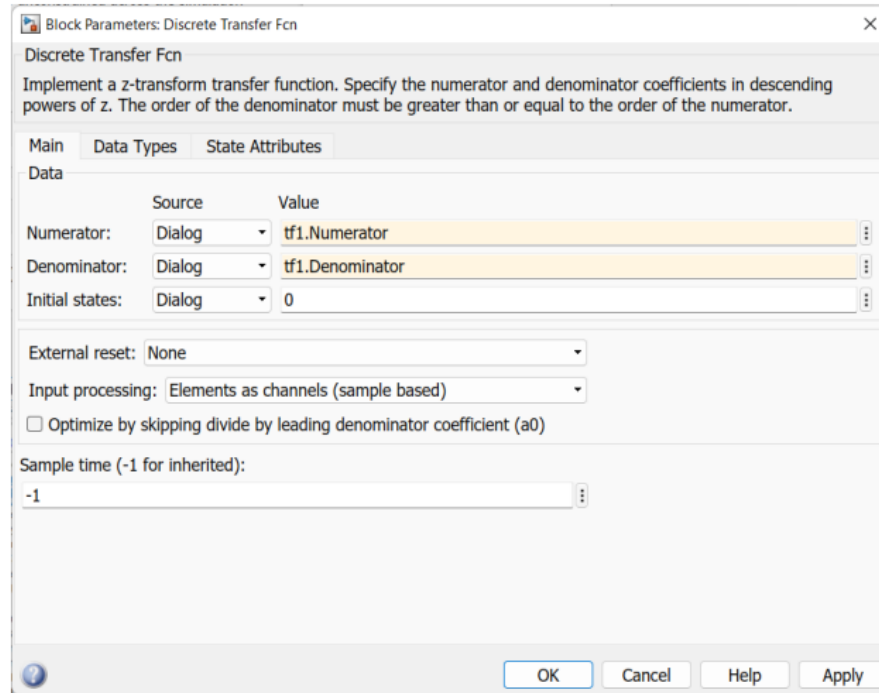
Parameterization:
  Number of poles: 1  Number of zeros: 1
  Number of free coefficients: 2
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
  Estimated using TFEST on time domain data "mydata1".
  Fit to estimation data: 89.51% (stability enforced)
  FPE: 0.1441, MSE: 0.1436

fx >>
```

15. เมื่อได้สมการ Transfer Function แล้ว นักศึกษาสามารถนำสัญญาณที่ได้จาก Transfer Function มาเปรียบเทียบกับสัญญาณจาก Encoder ได้ใน Validate Subsystem โดยมีขั้นตอนดังนี้

- เปลี่ยนค่า Numerator และ Denominator ภายใน Transfer Fcn block ให้ตรงตามสมการ Transfer Function ที่ได้จาก System Identification



หมายเหตุ นักศึกษาต้องเปลี่ยนตัวแปร “tf1” ให้ตรงตามชื่อ Transfer Function ใน workspace

LAB Report -> ไม่มี Format ให้ครับ

เกณฑ์การให้คะแนน

1. สามารถแสดงวิธีทำในการหาค่าพารามิเตอร์สำหรับการจำลองมอเตอร์ได้อย่างถูกต้อง
2. สามารถอธิบายแนวคิดในการเลือก Input และการ Validation ได้อย่างถูกต้อง
3. สามารถวิเคราะห์ผลการทดลองและอธิบายถึงความเป็น Non-Linear ได้อย่างถูกต้อง