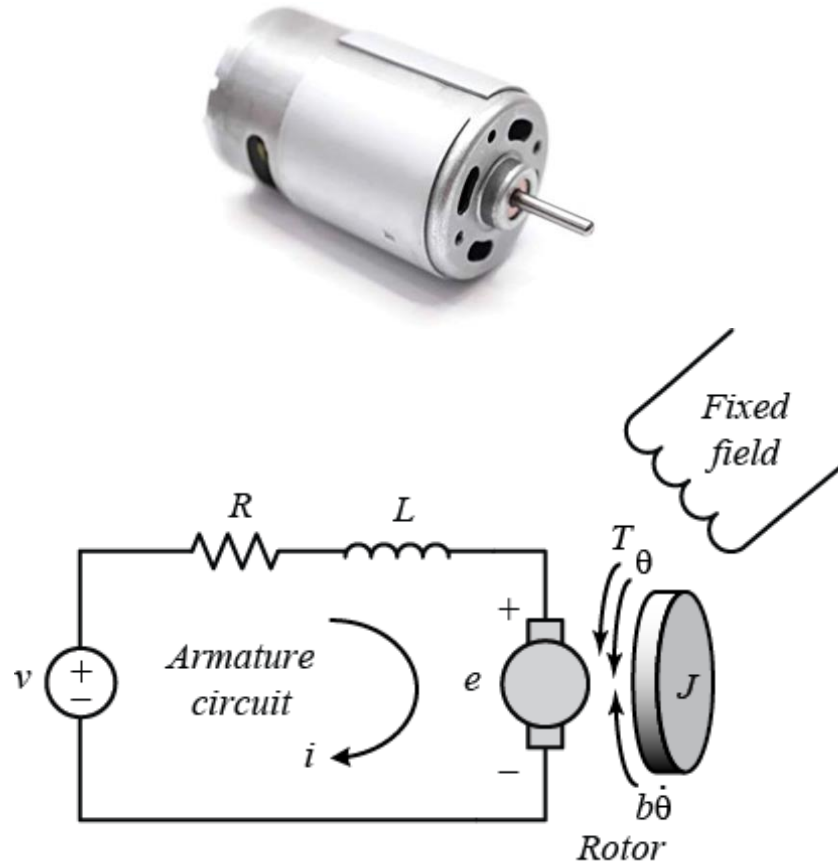


# System Identification

RMX : FRA231

Kitti Thamrongaphichartkul

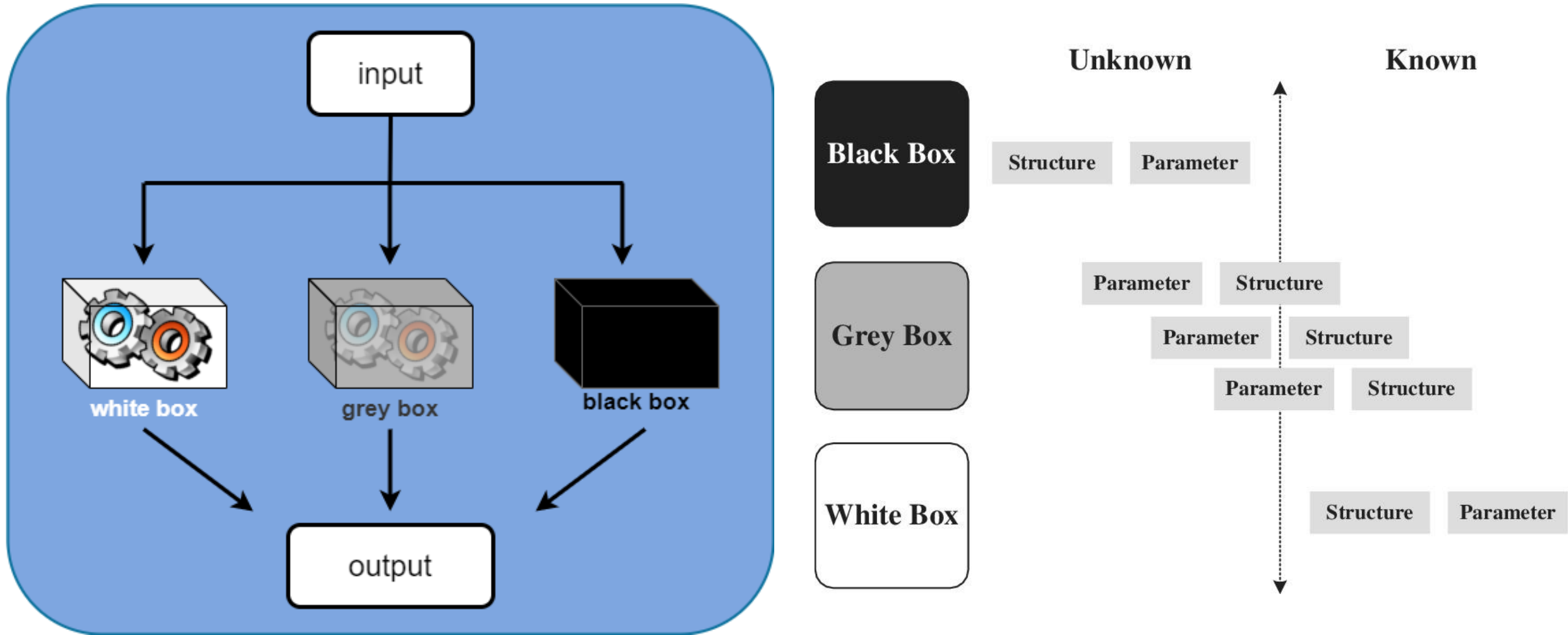
# DC Motor Speed: System Modeling



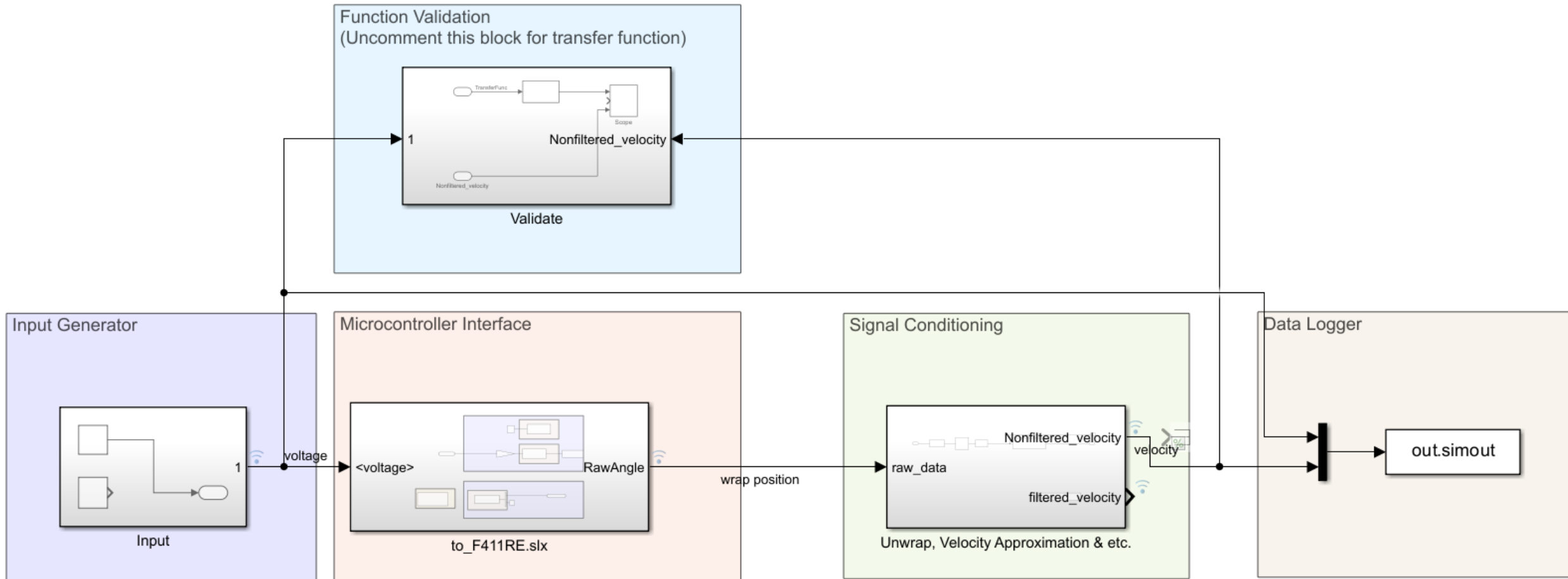
The physical parameters for our example are:

(J)	moment of inertia of the rotor	$0.01 \text{ kg.m}^2$
(b)	motor viscous friction constant	$0.1 \text{ N.m.s}$
(Ke)	electromotive force constant	$0.01 \text{ V/rad/sec}$
(Kt)	motor torque constant	$0.01 \text{ N.m/Amp}$
(R)	electric resistance	$1 \text{ Ohm}$
(L)	electric inductance	$0.5 \text{ H}$

# Black Box vs Grey Box vs White Box

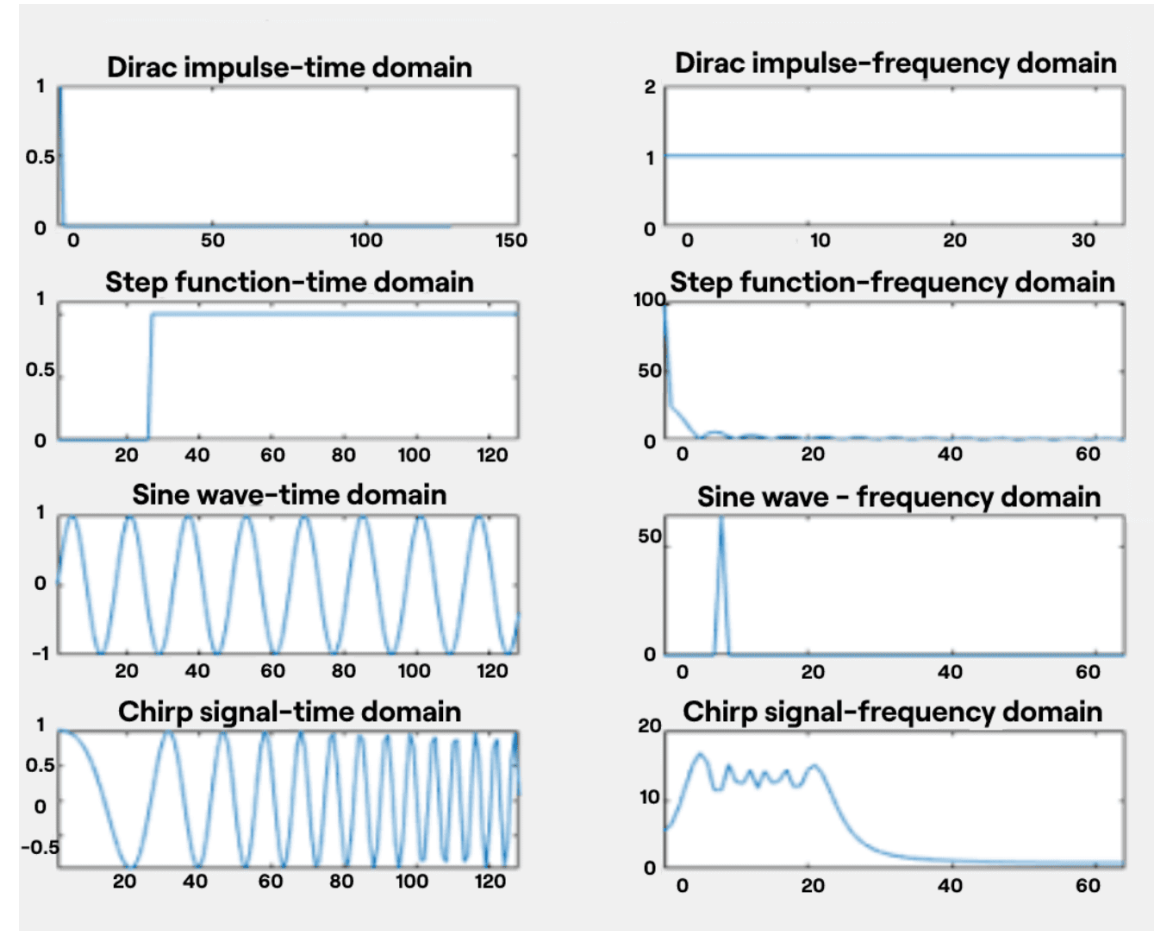
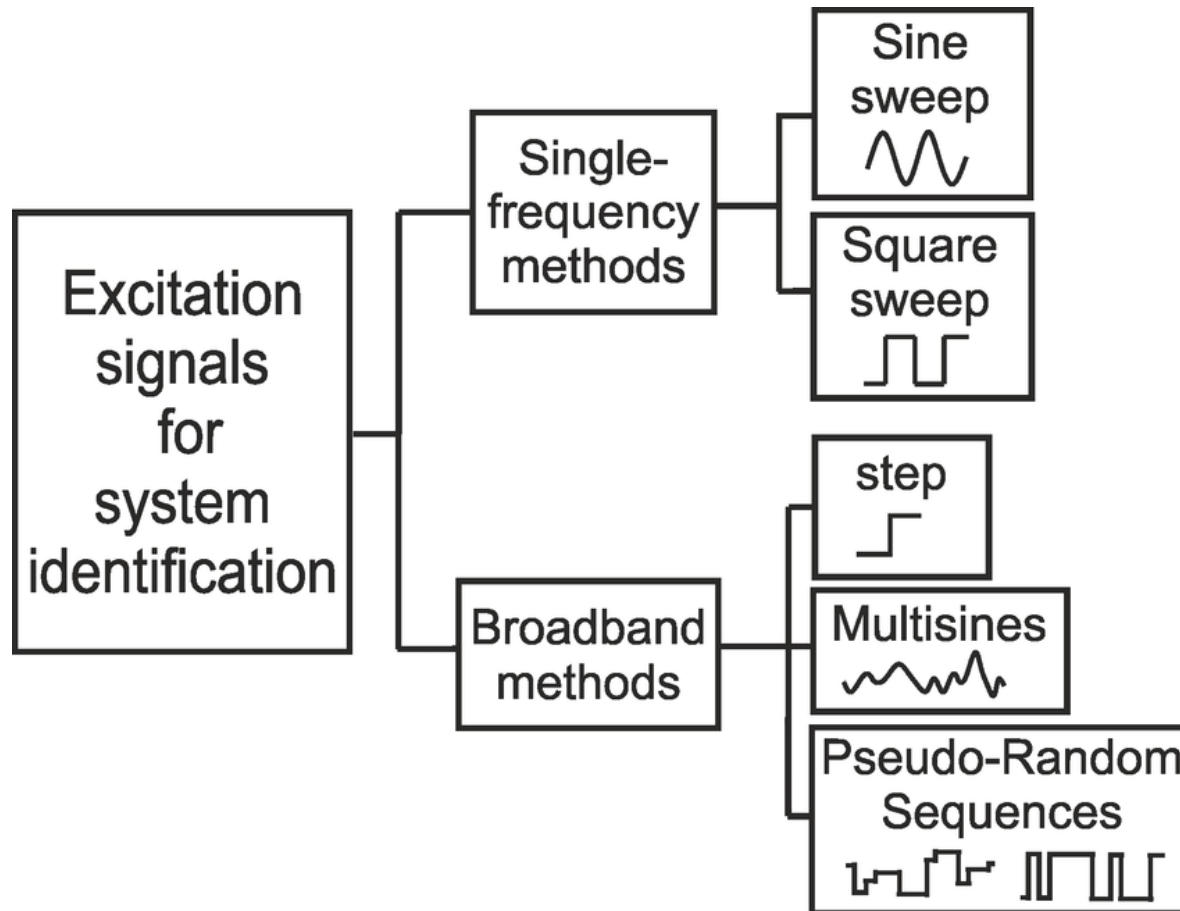


# System Identification Diagram

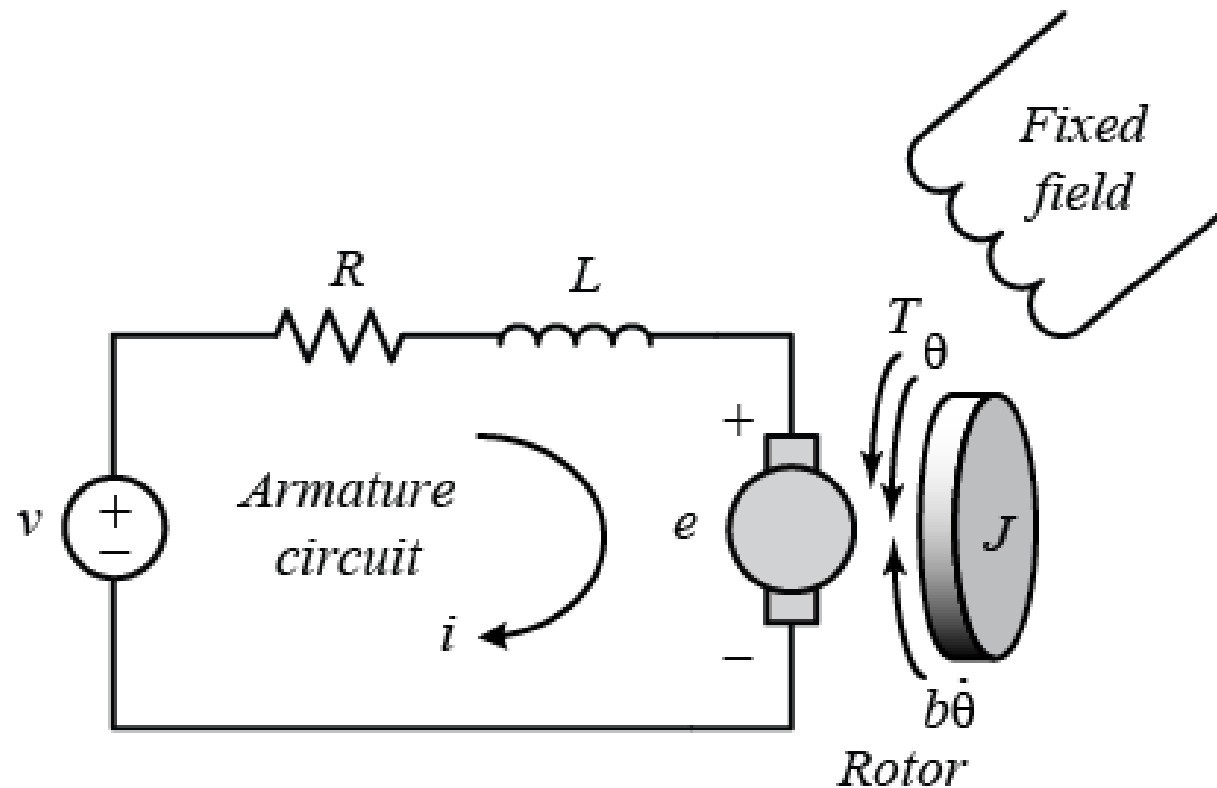


**\*\*Reminder\*\* check your com port in block "to\_f411RE.slx" before Run the model!!**

# Input Generator



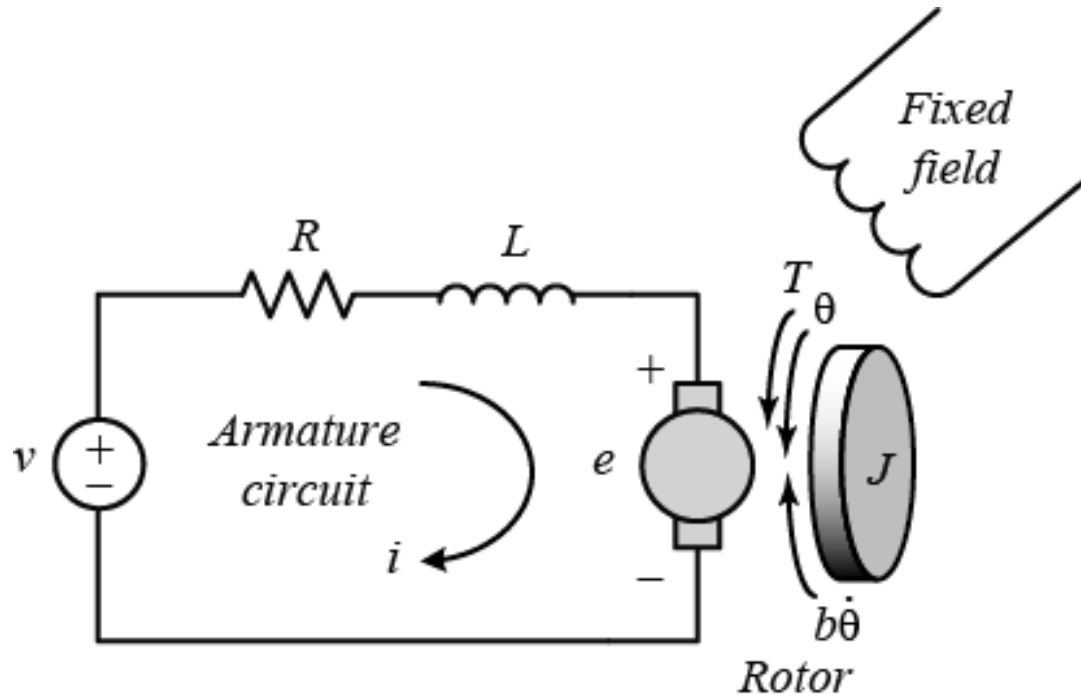
# DC Motor Speed: System Modeling



The physical parameters for our example are:

(J)	moment of inertia of the rotor	$0.01 \text{ kg.m}^2$
(b)	motor viscous friction constant	$0.1 \text{ N.m.s}$
(Ke)	electromotive force constant	$0.01 \text{ V/rad/sec}$
(Kt)	motor torque constant	$0.01 \text{ N.m/Amp}$
(R)	electric resistance	$1 \text{ Ohm}$
(L)	electric inductance	$0.5 \text{ H}$

# DC Motor Speed: System Modeling



## System equations

$$T = K_t i$$

$$e = K_e \dot{\theta}$$

$$L \frac{di}{dt} + Ri = V - K \dot{\theta}$$

$$J \ddot{\theta} + b \dot{\theta} = K i$$

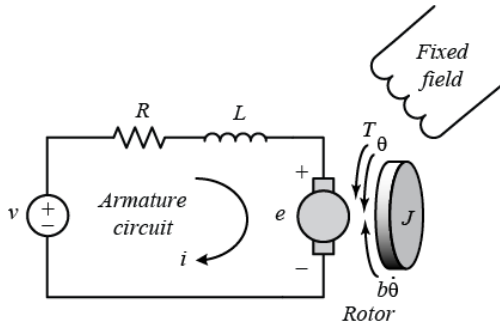
## Transfer Function

$$P(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad \left[ \frac{\text{rad/sec}}{V} \right]$$

## State Space

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

# DC Motor Speed: System Modeling



## System equations

$$T = K_t i$$

$$e = K_e \dot{\theta}$$

$$L \frac{di}{dt} + Ri = V - K \dot{\theta}$$

$$J \ddot{\theta} + b \dot{\theta} = K i$$

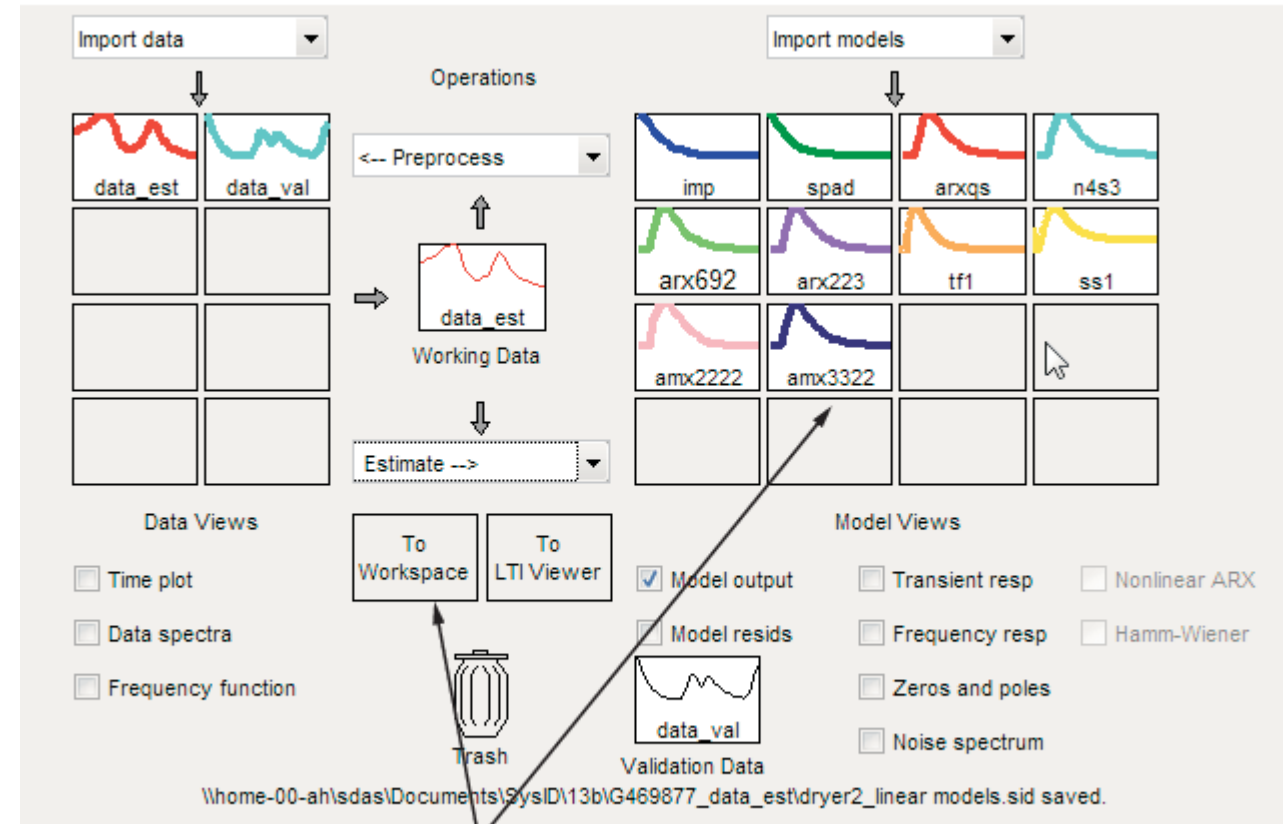
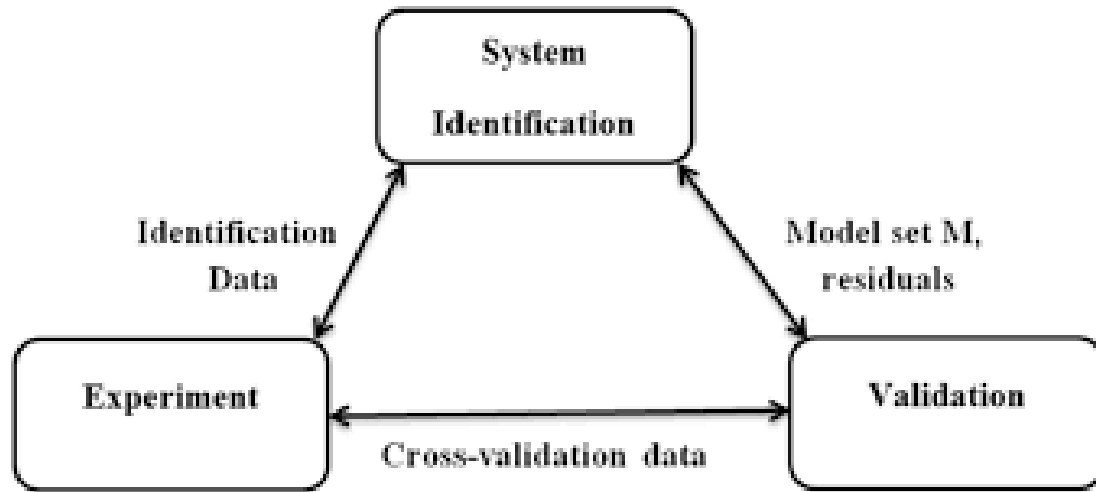
## State Space

Ans

$$\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \quad y = [1 \quad 0] \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$



# Validation Process

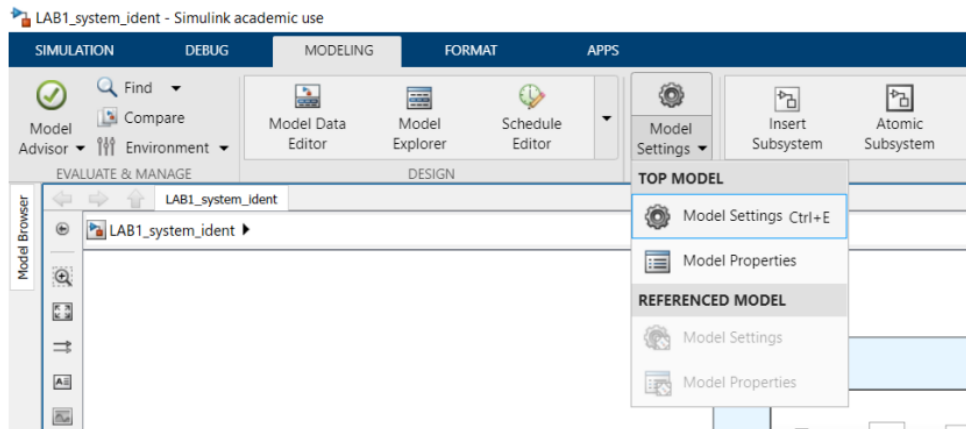


Drag and drop model to Workspace

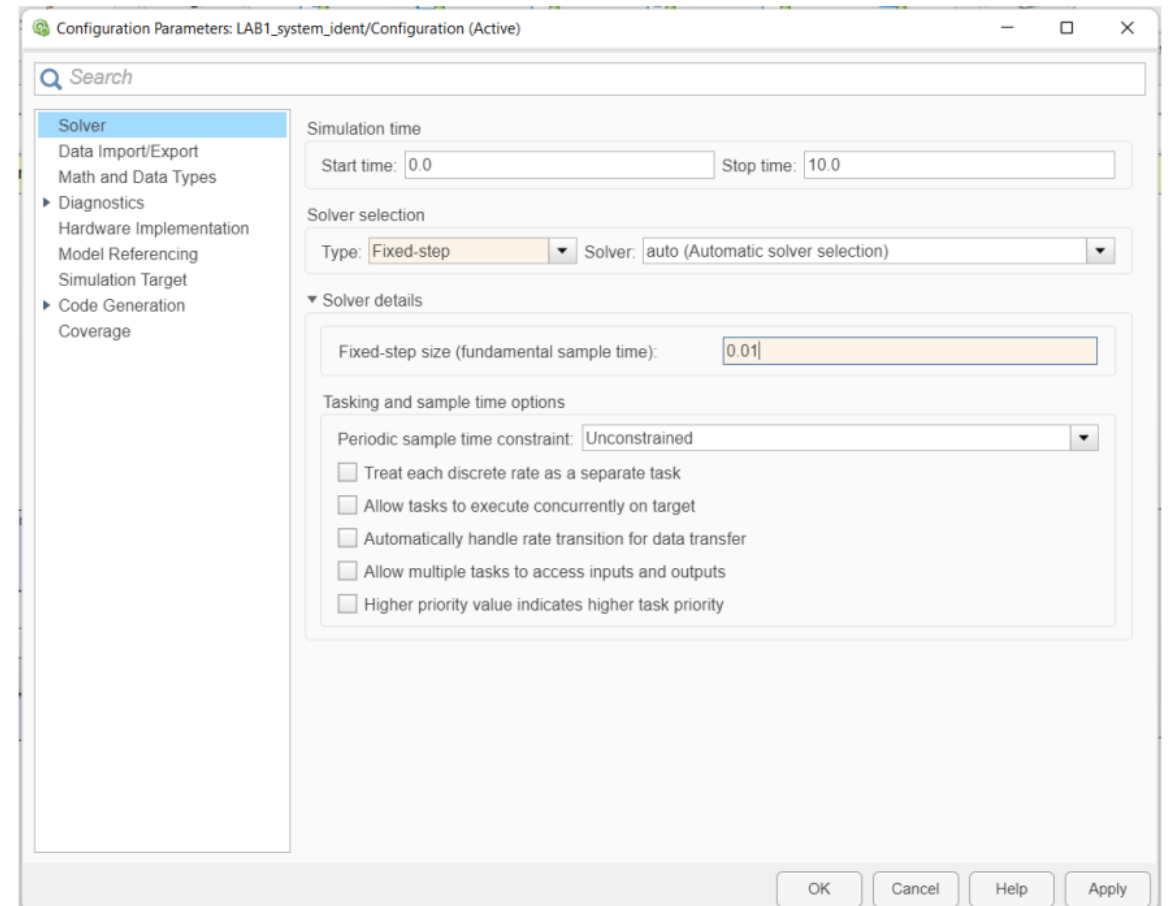
# Instruction

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



# Instruction

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 (สามารถใช้สัญญาณรูปแบบอื่นที่นอกเหนือจากตัวอย่างได้)

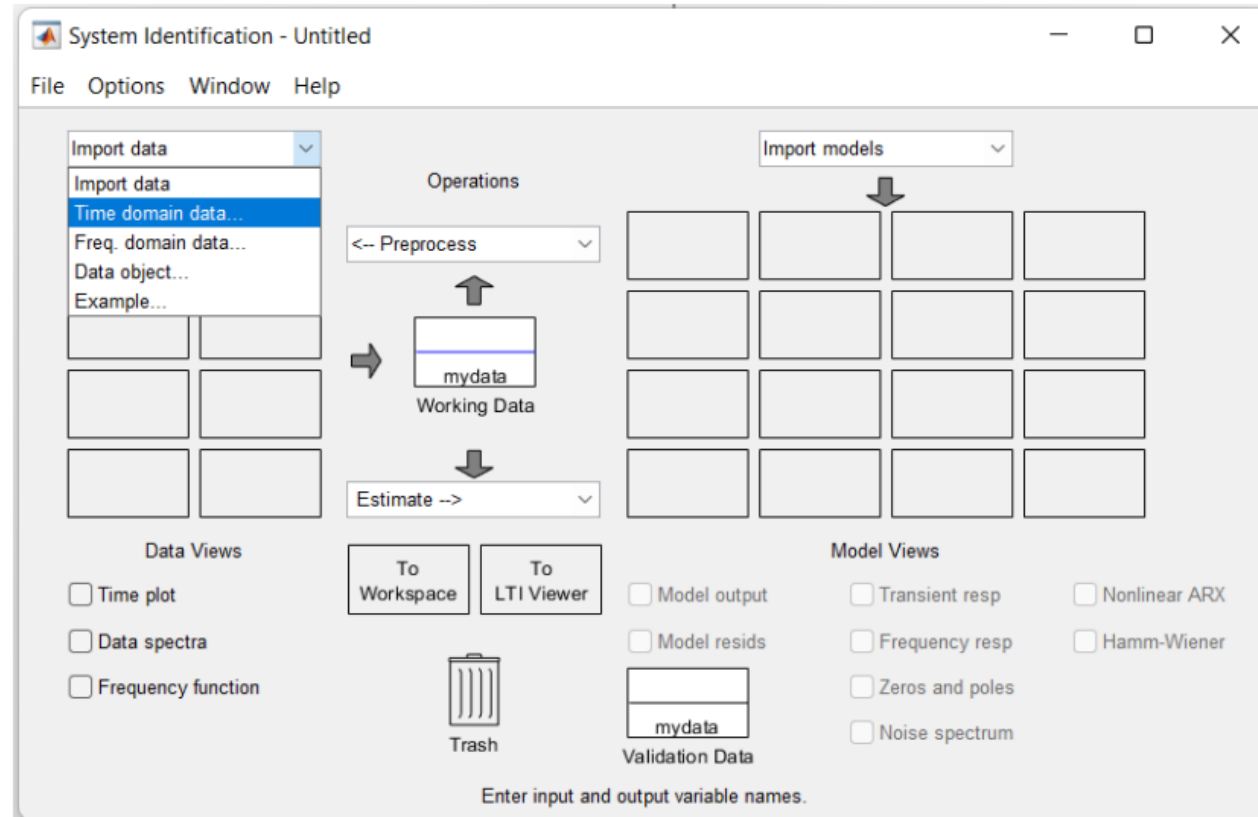
6. กด Run เพื่อเก็บค่าสัญญาณไปยัง Workspace
7. พิมพ์คำสั่ง "ident" ในหน้าต่าง Command Window

```
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 >>
```

# Instruction

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



# Instruction

9. กรอกค่าของ workspace variable ลงไป จากนั้นกดปุ่ม Import

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

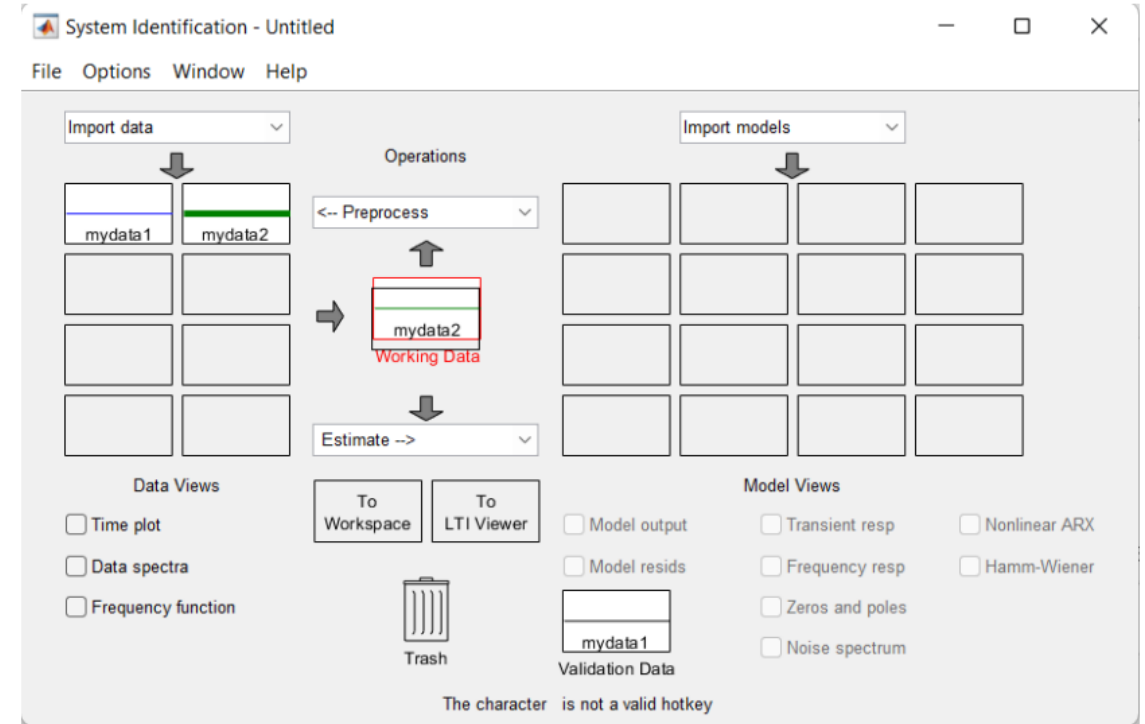
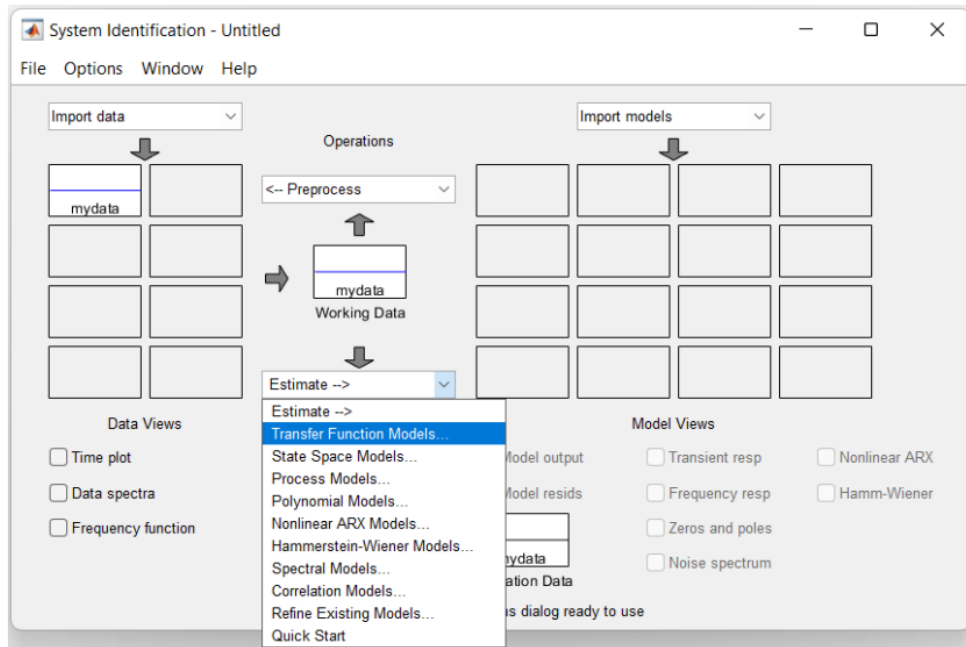
- Data Format for Signals:** Time Domain Signals
- Workspace Variable:**
  - Input: out.simout.Data(:,1)
  - Output: out.simout.Data(:,2)
- Data Information:**
  - Data Name: mydata
  - Start Time: 0
  - Sample time: 0.01
- Buttons:** Import, Reset, Close, Help

- 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

# Instruction

10. คลิกที่หัวข้อ Estimate > Transfer Function Model

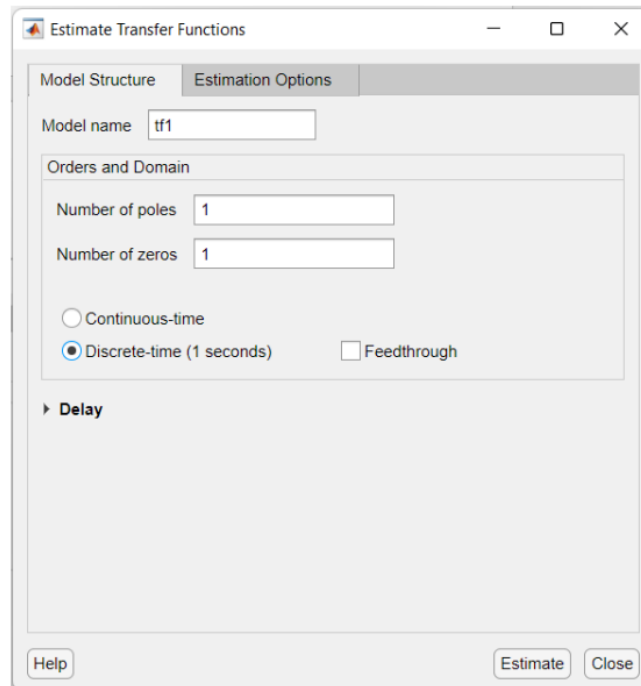


หมายเหตุ กรณีที่มี Data มากกว่า 1 ชุด นักศึกษาต้องลาก Block Data ที่ต้องการ Estimate ไปยัง Working Data

# Instruction

## 11. ประมาณค่า Transfer Function ของมอเตอร์

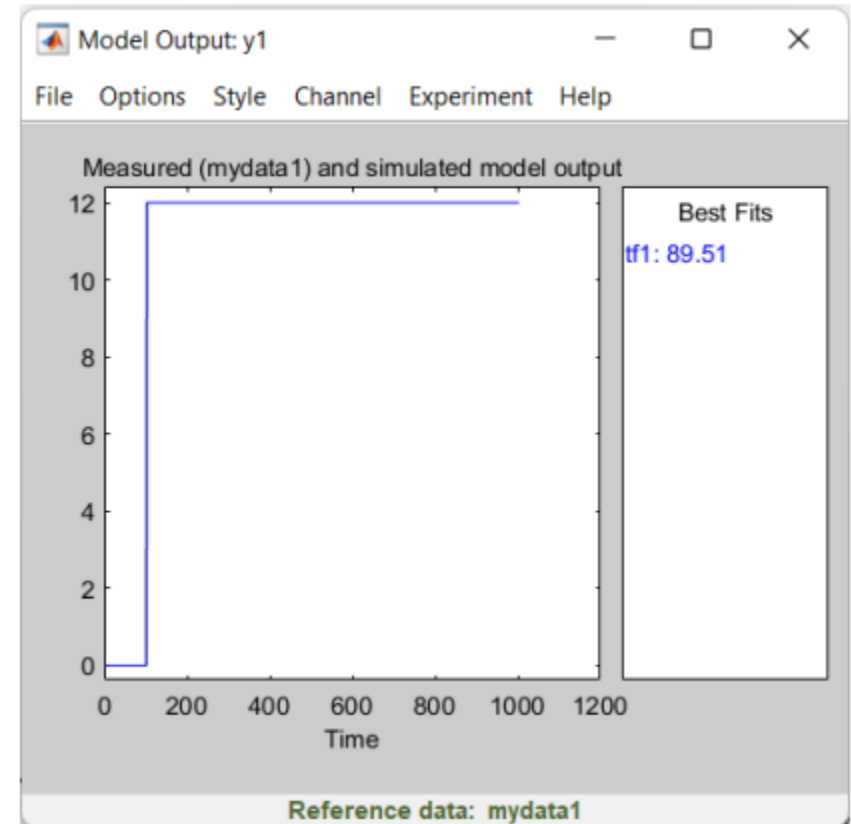
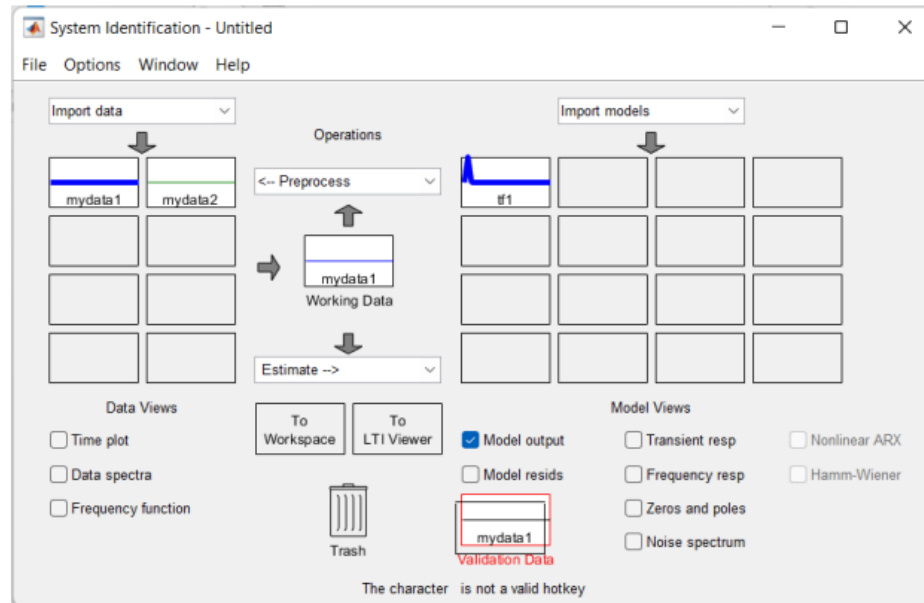
- กำหนดจำนวนของ pole และ zero ให้ถูกต้องตามสมการมอเตอร์ที่ได้จากการคำนวณ โดยต้องคำนวณเป็น Discrete-Time (Domain “z”)
- ~~เลือก Discrete Time (0.01 seconds)~~
- กดปุ่ม Estimate



# Instruction

12. เมื่อได้ Transfer Function แล้วนักศึกษาสามารถตรวจสอบความถูกต้อง Transfer Function ที่ได้กับชุดข้อมูลแต่ละชุดได้ โดย

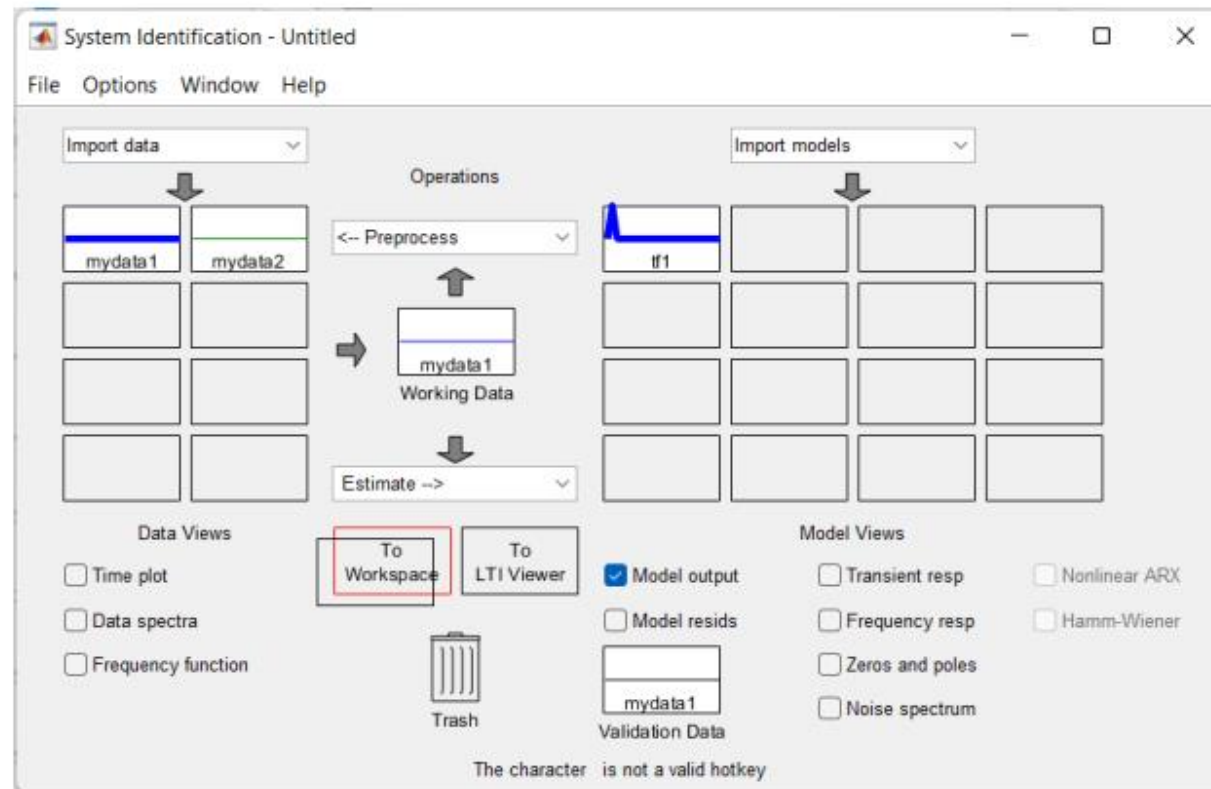
- ลาก Block Data ที่ต้องการ validate ไปยัง Validation Data
- กดปุ่ม Model Output ซึ่งจะแสดงค่าความถูกต้องระหว่าง Model ที่ถูกประมาณค่ากับค่าสัญญาณจริง





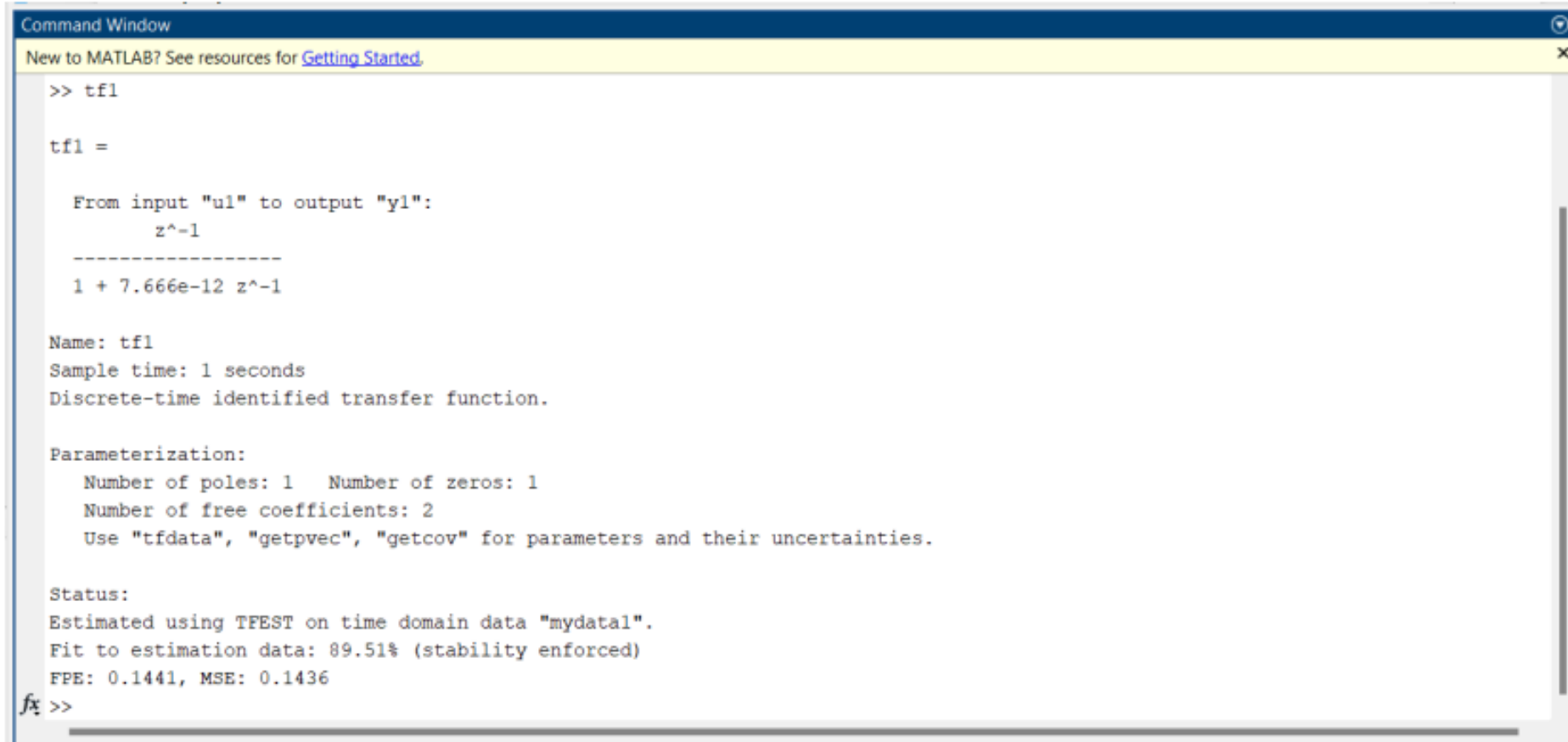
# Instruction

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



# Instruction

## 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 >>
```

# Topic

**System Identification**

->  
->

**Simulation**  
**Hardware**