CONTROLA

# Dynamics and Control for Quadrotor installed with CMGs.

2016 / Day Three

Kim. Young-Ouk.

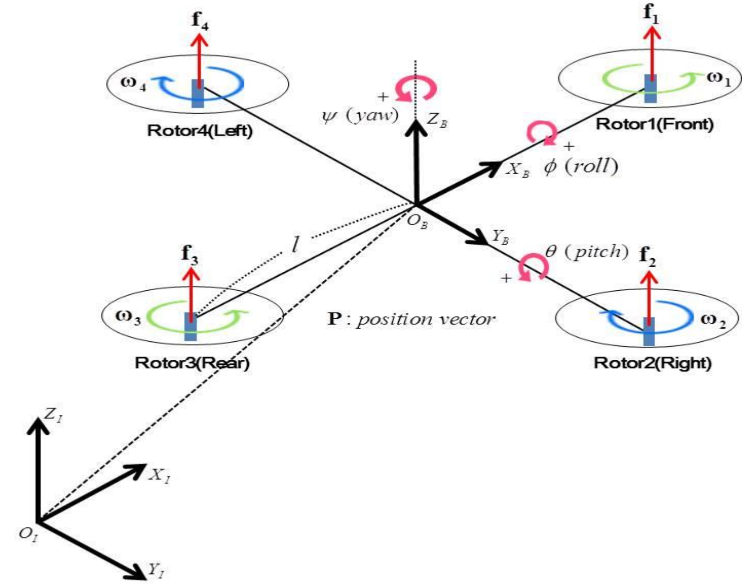
# Contents

# 1. 쿼드로터 쿼터니언 피드백 제어.

## 1. Quadrotor Dynamics.

### 1) Quadrotor Fright Principle.

$$\tau_\phi = U_1 = Lk_t(\omega_4^{\,2} - \omega_2^{\,2})$$
$$\tau_\phi = U_2 = Lk_t(\omega_1^{\,2} - \omega_3^{\,2})$$
$$\tau_\psi = U_3 = d(\omega_2^{\,2} + \omega_4^{\,2} - \omega_1^{\,2} - \omega_3^{\,2})$$
$$T_B = U_4 = k_t(\omega_1^{\,2} + \omega_2^{\,2} + \omega_3^{\,2} + \omega_4^{\,2})$$



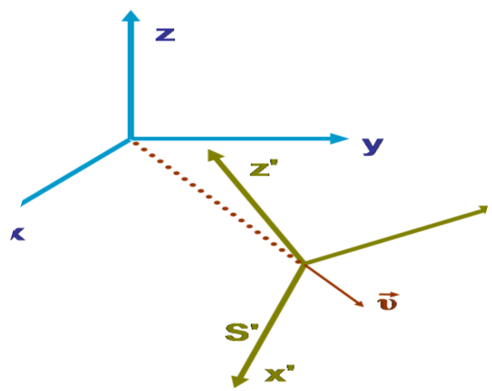

### 2) Quadrotor Dynamics.

$$\mathbf{V}_I = R_B^I \mathbf{V}_B$$
$$\dot{\mathbf{V}}_B = (R_I^B \mathbf{F}_{G,I} + \mathbf{T}_{T,B} + \mathbf{F}_{A,B} + \Delta\mathbf{F}_{D,B})/m - (\mathbf{\Omega}_B \times \mathbf{V}_B)$$
$$\mathbf{\Omega}_I = N_B^I \mathbf{\Omega}_B$$
$$\dot{\mathbf{\Omega}}_B = J^{-1}\{\boldsymbol{\tau}_{\tau,B} - (\mathbf{\Omega}_B \times J\mathbf{\Omega}_B)\}$$

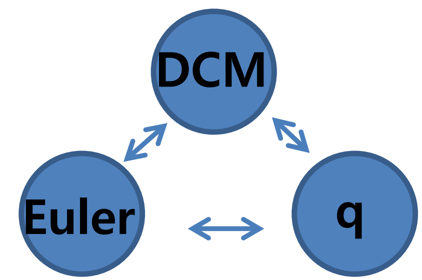
# 1. 쿼드로터 쿼터니언 피드백 제어.

## 2. Quaternion.

### 1) Attitude Represent.

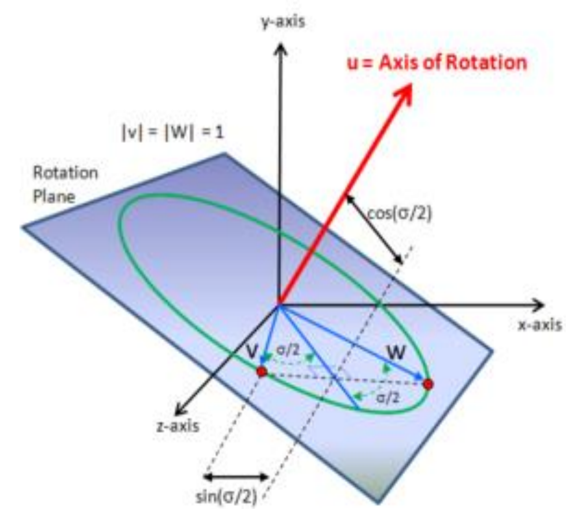



- Direction Cosine Matrix
- Euler Angle
- Quaternion




### 2) Quaternion.

- 정의 : 두 자세간의 기동은 한번의 회전으로 변환할 수 있다.
  (오일러 회전이론, 변수 : 4개 (오일러 각, 오일러 축))
- 계산과정이 간단하여 탑재 실시간 계산에 적합.
- 큰 자세각에서 특이점 없이 모든 자세 표현가능.



# 1. 쿼드로터 쿼터니언 피드백 제어.

## 2. Quaternion.

### 3) Quaternion definition.

$$\mathbf{q} = [q_0, \mathbf{q_v}]^T$$

$$\mathbf{q}^* = [q_0, -\mathbf{q_v}]^T$$

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 p_0 - \mathbf{q_v} \cdot \mathbf{p_v} \\ q_0 p_v + q_0 \mathbf{p_v} + p_0 \mathbf{q_v} + \mathbf{q_v} \times \mathbf{p_v} \end{bmatrix}$$

### 4) Quaternion Rotation.

$$\mathbf{v_2} = \mathbf{q} \otimes \mathbf{v_1} \otimes \mathbf{q}^*$$

$$\mathbf{v_2} = \mathbf{R_1^2}(q)\mathbf{v_1}$$

$$\begin{aligned} \mathbf{R_1^2}(q) &= (q_0{}^2 - \mathbf{q_v}{}^T \mathbf{q_v})\mathbf{I} + 2\mathbf{q_v}\mathbf{q_v}{}^T + 2q_0[q]^\times \\ &= \mathbf{vv} + \cos\theta(\mathbf{I} - \mathbf{vv}) + \sin\theta \mathbf{I} \times \mathbf{v} \\ &= \begin{bmatrix} 1 - 2q_2{}^2 - 2q_3{}^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_3{}^2 - 2q_1{}^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 1 - 2q_1{}^2 - 2q_2{}^2 \end{bmatrix} \end{aligned}$$

# 1. 쿼드로터 쿼터니언 피드백 제어.

## 3. Quaternion Feedback Control.

### 1) Quaternion error.

$$\mathbf{q} = \mathbf{q_e} \otimes \mathbf{q_d}$$

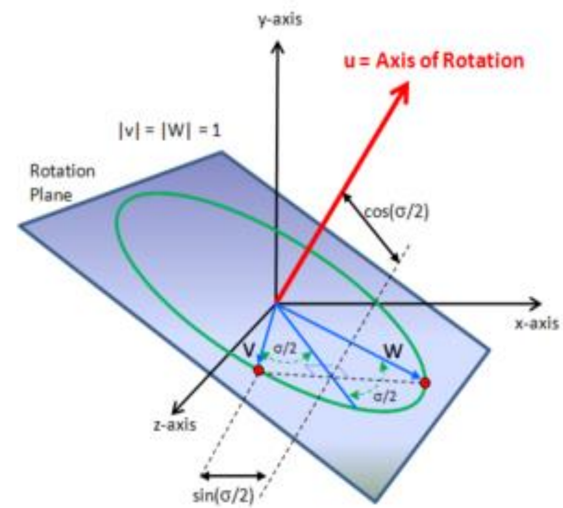
$$\mathbf{q}_e = \mathbf{q} \otimes \mathbf{q_d}^* = \begin{bmatrix} q_0 q_{0\mathbf{d}} - \mathbf{q}_v \cdot (-\mathbf{q}_{v\mathbf{d}}) \\ q_0(-\mathbf{q_{vd}}) + q_{0\mathbf{d}}\mathbf{q_v} + \mathbf{q}_v \times (-\mathbf{q}_{v\mathbf{d}}) \end{bmatrix}$$

### 2) Quaternion Kinematics.

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} -\mathbf{q}_v{}^T \boldsymbol{\omega} \\ (q_0 \mathbf{I} + \mathbf{S}(q))\boldsymbol{\omega} \end{bmatrix}$$

### 3) Quaternion Feedback control.

$$\mathbf{u} = -\mathbf{K}_p \mathbf{q_e} - \mathbf{K}_d \boldsymbol{\omega}$$

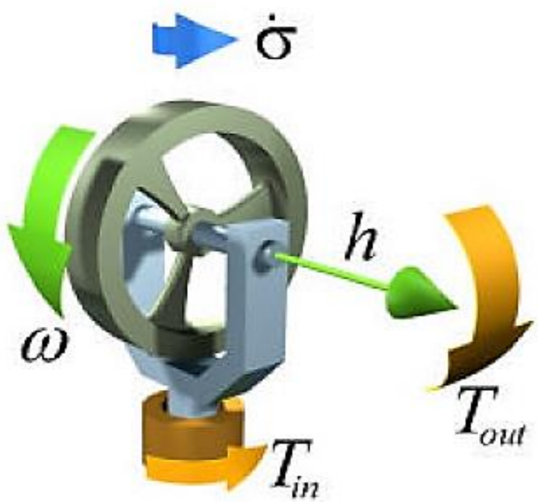


# Contents

# 2. CMG를 적용한 쿼드로터의 모델링.

## 1. CMG.

### 1) Control Moment Gyro Principle.




$$\vec{\boldsymbol{\tau}} = \dot{\vec{\boldsymbol{\delta}}} \times \vec{\mathbf{h}}$$

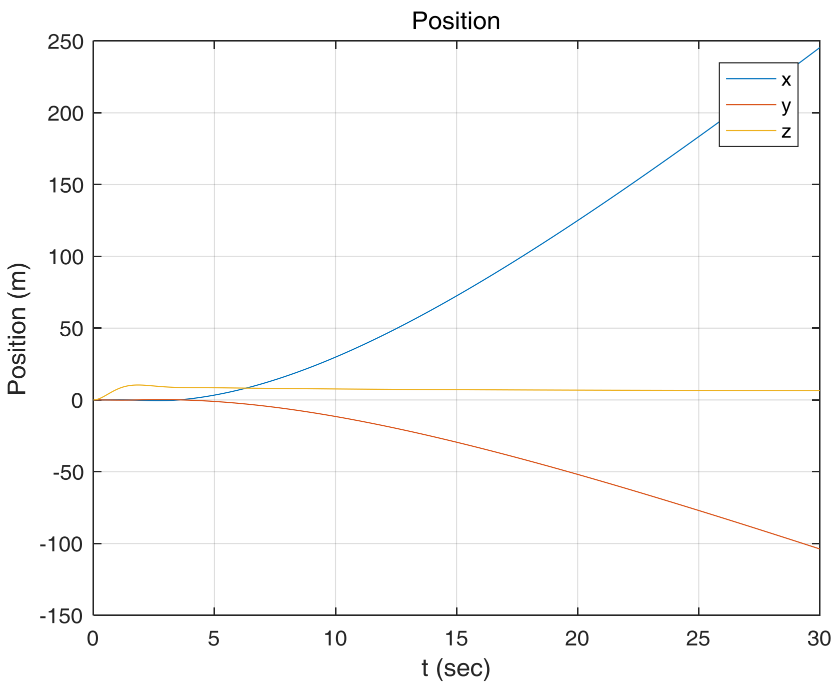
- CMG는 플라이휠(Flywheel)과 스핀모터(Spin Motor), 김벌모터(Gimbal Motor)로 구성.
- CMG는 모멘텀 교환원리를 이용한 구동기로 상대적으로 작은 조작에도 큰 토크를 얻을 수 있다는 장점이 있다.

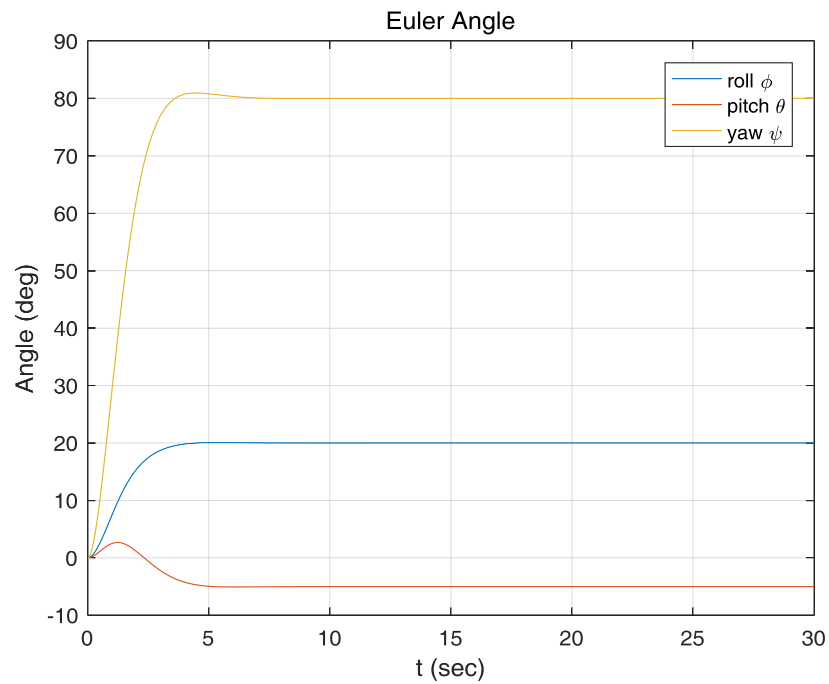
# 1. 쿼드로터 쿼터니언 피드백 제어.

## 3. Quaternion Feedback Control.

### 4) Simulation.

**Command** Altitude = 10m Attitude = 20, −5, 80 deg



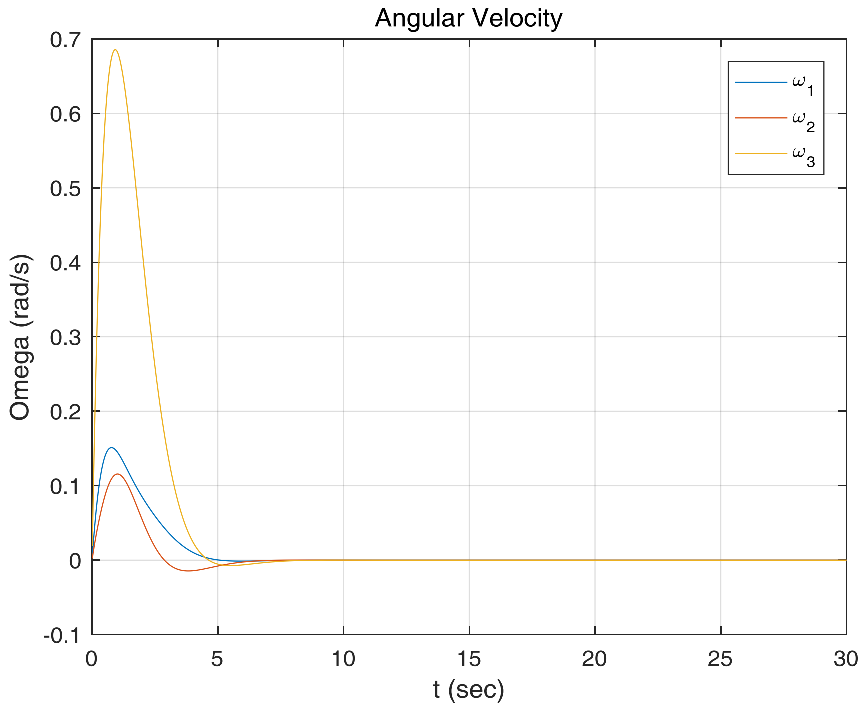



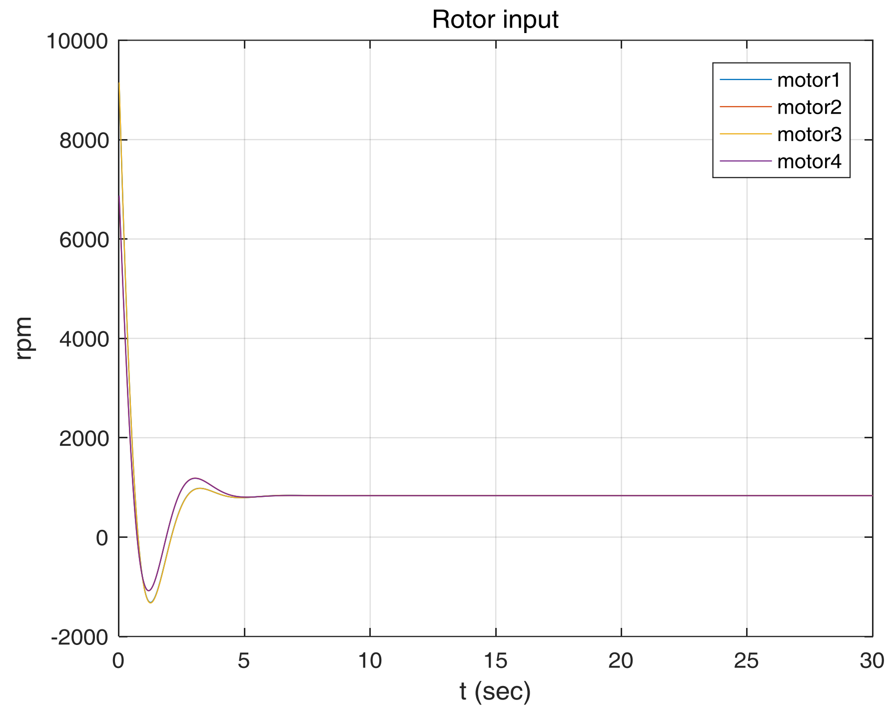
# 1. 쿼드로터 쿼터니언 피드백 제어.

## 3. Quaternion Feedback Control.

4) Simulation.

**Command** Altitude = 10m Attitude = 20, −5, 80 deg

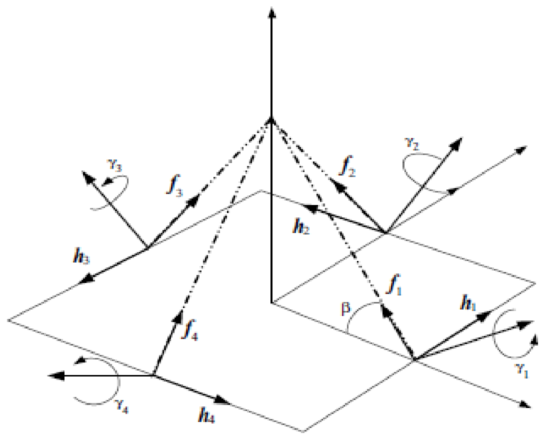


# 2. CMG를 적용한 쿼드로터의 모델링.

## 2. Pyramid type of 4-CMGs.

1) Configuration of Pyramid type of 4-CMGs.




2) Modeling of 4-CMG's pyramid type.

$$\begin{bmatrix} c\beta\cos\gamma_1 & \sin\gamma_2 & c\beta\cos\gamma_3 & \sin\gamma_4 \\ \sin\gamma_1 & c\beta\cos\gamma_2 & \sin\gamma_3 & c\beta\cos\gamma_4 \\ s\beta\cos\gamma_1 & s\beta\cos\gamma_2 & s\beta\cos\gamma_3 & s\beta\cos\gamma_4 \end{bmatrix}$$

# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

1) Quadrotors Dynamics Model.

Linear Motion

$$\mathbf{V}_I = R_B^I \mathbf{V}_B$$

$$\dot{\mathbf{V}}_B = (R_I^B \mathbf{F}_{G,I} + \mathbf{T}_{T,B} + \mathbf{F}_{A,B} + \Delta\mathbf{F}_{D,B}) / m - (\mathbf{\Omega}_B \times \mathbf{V}_B)$$

Rotational Motion

$$\mathbf{\Omega}_I = N_B^I \mathbf{\Omega}_B$$

$$\dot{\mathbf{\Omega}}_B = J^{-1}\{\boldsymbol{\tau}_{\tau,B} - (\mathbf{\Omega}_B \times J\mathbf{\Omega}_B)\}$$ + 4_CMGs.

CMG torque

$$\boxed{\vec{\boldsymbol{\tau}} = \vec{\dot{\boldsymbol{\delta}}} \times \vec{\mathbf{h}}}$$

# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

2) Modeling of quadrotor installed with 4-CMGs.

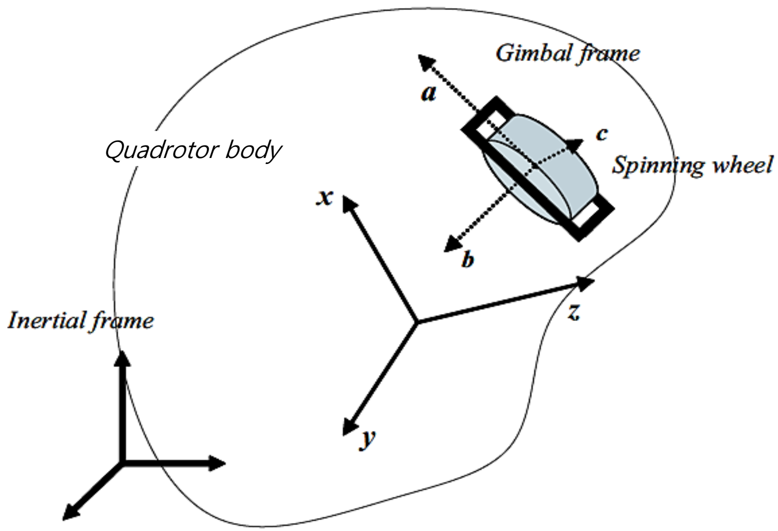



그림 1

$$C = [\,a\ b\ c\,]$$ (DCM)

$$\omega = C\omega_g$$ (Angular Velocity)

$$\dot{\gamma}_g = [\,\dot{\gamma}\ 0\ \ 0\,]^T$$ (gimbal rate)

$$\bar{\varpi}_g = [\,0\ \bar{\varpi}\ 0\,]^T$$ (spin rate)

- 쿼드로터와 CMG의 좌표계는 그림1과 같다.
- CMG인 김벌좌표계에서 쿼드로터의 동체좌표계로의 좌표변환은 좌표변환행렬 $C$를 따른다.
- 쿼드로터의 각속도는 김벌좌표계에서의 각속도와 좌표변환행렬의 곱으로 나타낼 수 있다.
- 김벌의 각속도와 회전휠의 각속도는 각각 $\dot{\gamma}_g$, $\bar{\varpi}_g$ 로 나타낸다.

# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

### 2) Modeling of quadrotor installed with 4-CMGs.

- Total Angular Momentum

$$h = h_s + h_g + h_d$$

$$h_s = J_s \omega$$

$$h_g = CI_g(\omega_g + \dot{\gamma}_g)$$

$$h_d = CI_d(\omega_g + \dot{\gamma}_g + \overline{\omega}_g)$$

- Euler's Equation of Motion

$$J\dot{\omega} + \omega \times J\omega = M$$

- Kinematic differential equation

$$\dot{C} = C\dot{\gamma}_g^{\times}$$

- 쿼드로터의 전체 각운동량은 쿼드로터와 김벌, 휠디스크가 생성하는 모멘텀의 총합이다.
- 오일러의 회전 운동방정식을 통해 쿼드로터의 운동방정식을 유도한다.
- 김벌이 시간에 따라 변하므로 김벌좌표계의 시간의 변화율을 계산해야 한다.

# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

2) Modeling of quadrotor installed with 4-CMGs.

- Dynamics of Multiple CMGs

$$J\dot{\omega} + \omega^{\times} J\omega + \omega^{\times} C_{\omega}(\gamma) h_{\omega}(\bar{\omega}) = -C_{\gamma}(\gamma) H_{\omega}(\bar{\omega}) \dot{\gamma}$$

$$\dot{C}_{\omega} = C_{\gamma} G, \quad \dot{C}_{\gamma} = -C_{\omega} G$$

$$C_{\omega}(\gamma) = [b_1 \; b_2 \; \ldots \; b_N]$$

$$C_{\gamma}(\gamma) = [c_1 \; c_2 \; \ldots \; c_N]$$

$$\gamma = [\gamma_1 \; \gamma_2 \; \ldots \; \gamma_N]^T$$

$$\bar{\omega} = [\bar{\omega}_1 \; \bar{\omega}_2 \; \ldots \; \bar{\omega}_N]^T$$

$$h_{\omega}(\bar{\omega}) = [I_{ab1}\bar{\omega}_1 \; I_{ab2}\bar{\omega}_2 \; \ldots \; I_{abN}\bar{\omega}_N]^T$$

$$I_{\omega} = \begin{bmatrix} I_{ab1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & I_{abN} \end{bmatrix} \quad H_{\omega}(\bar{\omega}) = \begin{bmatrix} h_{\omega 1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & h_{\omega N} \end{bmatrix}$$

$$G = \begin{bmatrix} \dot{\gamma}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \dot{\gamma}_N \end{bmatrix}$$

# 2. CMG를 적용한 쿼드로터의 모델링.

**3. Quadrotor modeling installed with pyramid type of 4-CMGs**

2) Modeling of quadrotor installed with 4-CMGs.

Linear Motion

$$\mathbf{V}_I = R_B^I \mathbf{V}_B$$

$$\dot{\mathbf{V}}_B = (R_I^B \mathbf{F}_{G,I} + \mathbf{T}_{T,B} + \mathbf{F}_{A,B} + \Delta\mathbf{F}_{D,B}) / m - (\mathbf{\Omega}_B \times \mathbf{V}_B)$$

Rotational Motion

$$\mathbf{\Omega}_I = N_B^I \mathbf{\Omega}_B$$

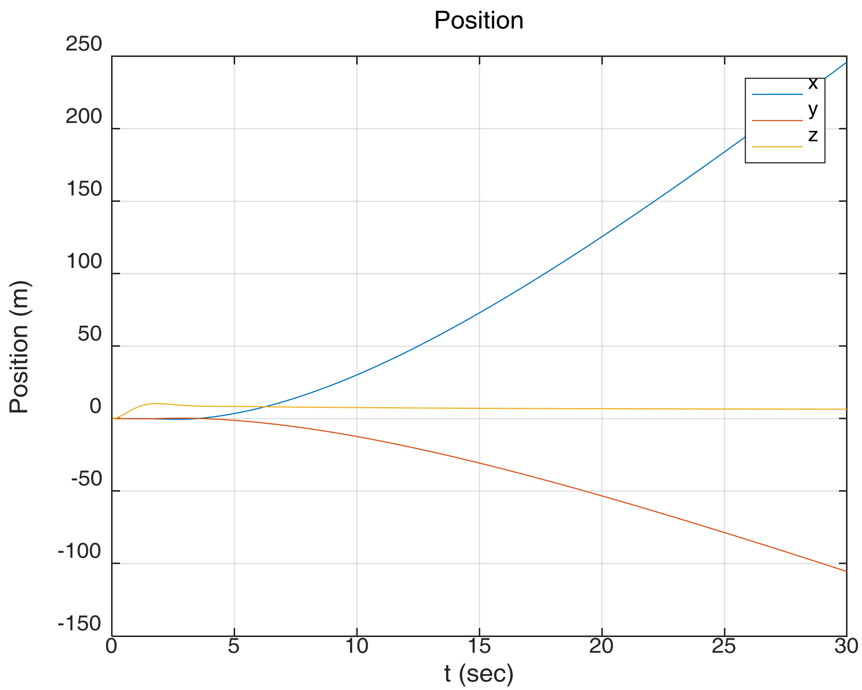
$$\dot{\mathbf{\Omega}}_B = J^{-1}\{\boldsymbol{\tau}_{\tau,B} - (\mathbf{\Omega}_B \times J\mathbf{\Omega}_B) \underbrace{- \mathbf{\Omega}_B^{\times} \mathbf{C}_\omega(\gamma)\mathbf{h}_\omega(\bar{\omega}) - \mathbf{C}_\gamma(\gamma)\mathbf{H}_\omega(\bar{\omega})\dot{\boldsymbol{\gamma}}}_{\text{4\_CMGs.}}\}$$

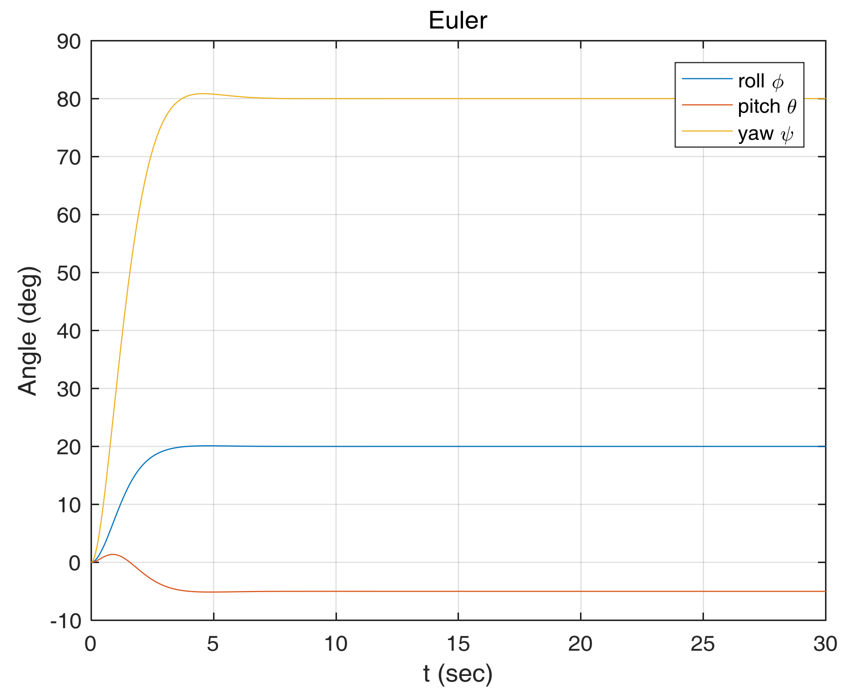
# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

### 3) Simulation

**Command** Altitude = 10m Attitude = 20, −5, 80 deg



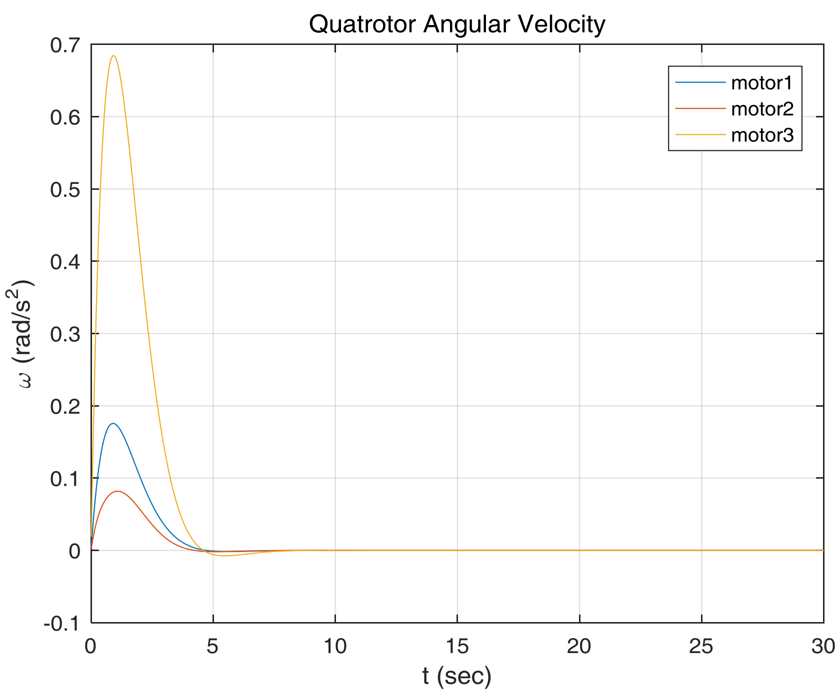



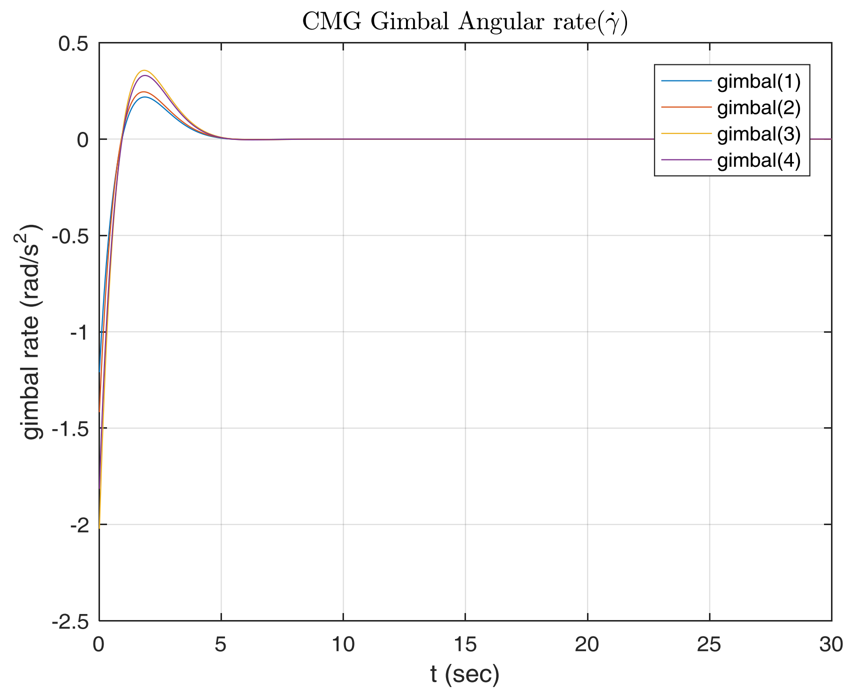
# 2. CMG를 적용한 쿼드로터의 모델링.

## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

### 3) Simulation

**Command** Altitude = 10m Attitude = 20, −5, 80 deg



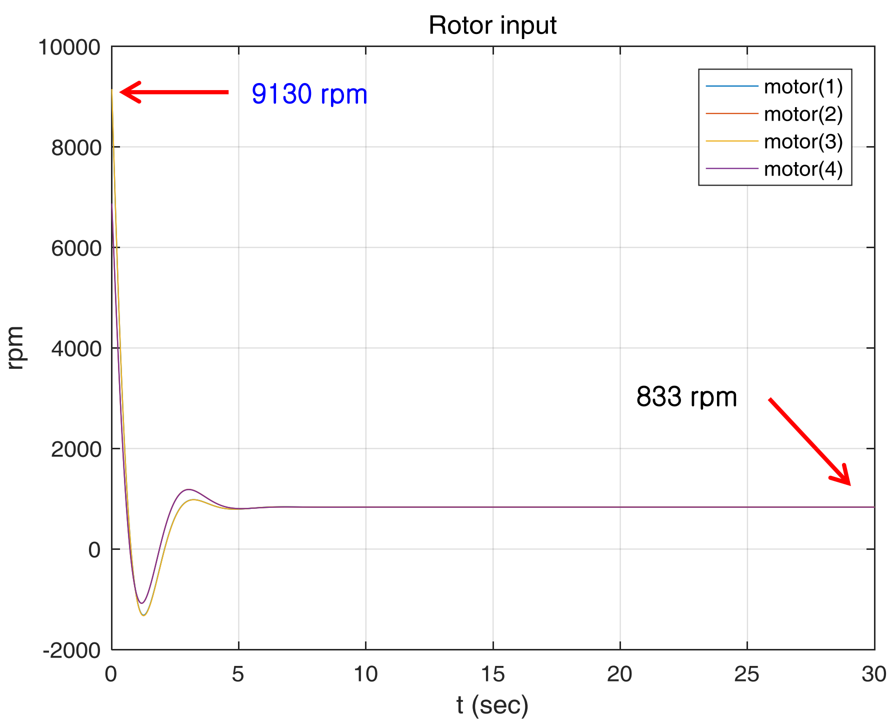



# 2. CMG를 적용한 쿼드로터의 모델링.

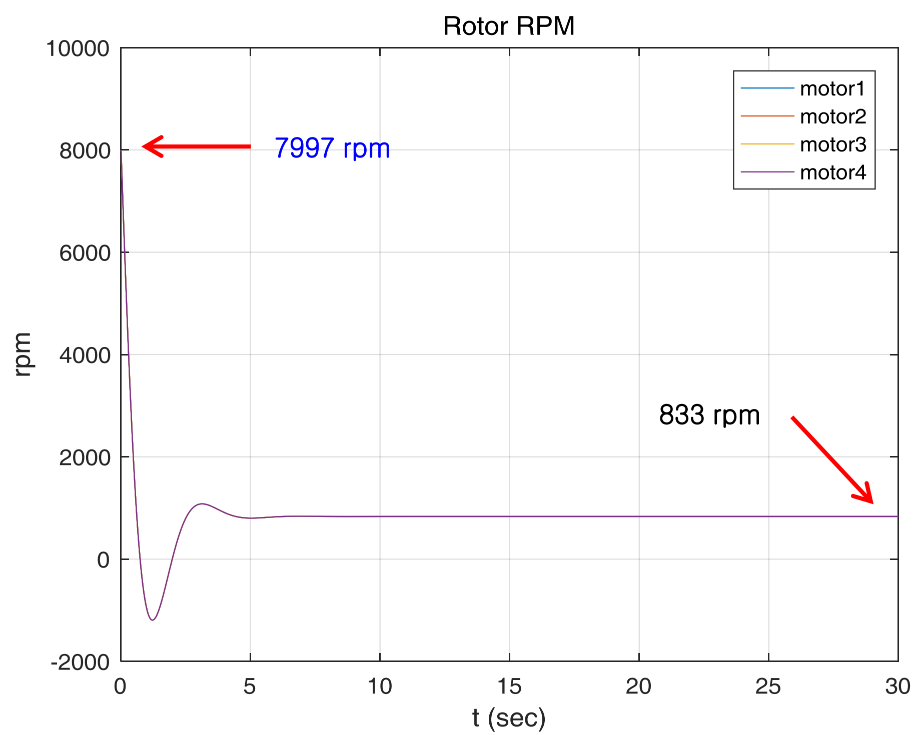
## 3. Quadrotor modeling installed with pyramid type of 4-CMGs

### 3) Simulation

**Quadrotor**




**Quadrotor + 4CMGs**



# Contents

# 3. 구동기의 최적 작동 법칙.

## 1. Least Square Method.

***Minimize Cost Function*** $J$

***Constrain Function*** $C$

$$\begin{cases} J : \|\mathbf{x}\|^2 \\ C : \mathbf{Ax} - \mathbf{y} = 0 \end{cases}$$

$$\mathbf{x} = [x_1 \ldots x_n]^T$$

$$\mathbf{A} = m \times n \quad (m > n)$$

$$\rightarrow \text{fat\_matrix.}$$

$$where,\ J : \|\mathbf{x}\|^2 = x_1^2 + x_2^2 + \cdots + x_n^2 = \mathbf{x}^T\mathbf{x}$$

① Using Indirect Method : Lagrange Multipliers.

$$\mathrm{L}(x, \lambda) : \ \mathbf{x}^T\mathbf{x} + \boldsymbol{\lambda}(\mathbf{Ax} - \mathbf{y})$$

② Optimality condition.

$$\nabla_x \mathrm{L} = \frac{\partial \mathrm{L}}{\partial x} : \ 2\mathbf{x}^T + \boldsymbol{\lambda}\mathbf{A} = 0 \qquad \nabla_\lambda \mathrm{L} = \frac{\partial \mathrm{L}}{\partial \lambda} : \ \mathbf{Ax} - \mathbf{y} = 0$$

# 3. 구동기의 최적 작동 법칙.

## 1. Least Square Method.

***Minimize Cost Function*** $J$

***Constrain Function*** $C$

$$\begin{cases} J : \|x\|^2 \\ C : Ax - y = 0 \end{cases}$$

③ Arrange.

$$2x^T + \lambda A = 0$$

$$Ax - y = 0$$

$$x = -\frac{1}{2}A^T\lambda$$

$$\lambda = -2(AA^T)^{-1}y$$

④ Solution.

$$x_{LSM} = A^{-1}(AA^T)^{-1}y$$

← Optimal Value.

# 3. 구동기의 최적 작동 법칙.

## 2. Weighted Least Square Method.

***Minimize Cost Function*** $J$

***Constrain Function*** $C$

$$\begin{cases} J : \mathrm{w}_1 \|\mathbf{a}\|^2 + \mathrm{w}_2 \|\mathbf{b}\|^2 \\ C_1 : \mathbf{A}_1\mathbf{a} - \mathbf{c} = 0, \quad C_2 : \mathbf{A}_2\mathbf{b} - \mathbf{d} = 0 \end{cases}$$

---

Let, $\mathbf{x} = [a_1, \cdots, a_n, b_1, \cdots, b_n]^T$

and, $$[x_1, \cdots, x_n] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} w_{ii} x_i x_i = \mathrm{w}_1 \|\mathbf{a}\|^2 + \mathrm{w}_2 \|\mathbf{b}\|^2$$

Hence, $J : \mathbf{x}^T \mathbf{W} \mathbf{x}$

and, $C : \mathbf{A}\mathbf{x} = \mathbf{y}$

$$\left( \begin{array}{c} \mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 \\ \mathbf{y} = \mathbf{c} + \mathbf{d} \end{array} \right)$$

# 3. 구동기의 최적 작동 법칙.

## 2. Weighted Least Square Method.

***Minimize Cost Function*** $J$

***Constrain Function*** $C$

$$\begin{cases} J : \mathrm{w}_1 \|\mathrm{a}\|^2 + \mathrm{w}_2 \|\mathrm{b}\|^2 \\ C_1 : \mathrm{A}_1\mathrm{a} - \mathrm{c} = 0, \quad C_2 : \mathrm{A}_2\mathrm{b} - \mathrm{d} = 0 \end{cases}$$

① Using Indirect Method : Lagrange Multipliers.

$$\mathrm{L}(x, \lambda) : \mathrm{x}^T \mathrm{W} \mathrm{x} + \lambda(\mathrm{A}\mathrm{x} - \mathrm{y})$$

② Optimality condition.

$$\nabla_x \mathrm{L} = \frac{\partial \mathrm{L}}{\partial x} : \ 2\mathrm{x}^T \mathrm{W} + \lambda \mathrm{A} = 0 \qquad \nabla_\lambda \mathrm{L} = \frac{\partial \mathrm{L}}{\partial \lambda} : \ \mathrm{A}\mathrm{x} - \mathrm{y} = 0$$

③ Solution.

$$\mathrm{x}_{WLSM} = \mathrm{W}^{-1}\mathrm{A}^{-1}(\mathrm{A}\mathrm{W}^{-1}\mathrm{A}^T)^{-1}\mathrm{y}$$

← Weighted Optimal Value.

# 3. 구동기의 최적 작동 법칙.

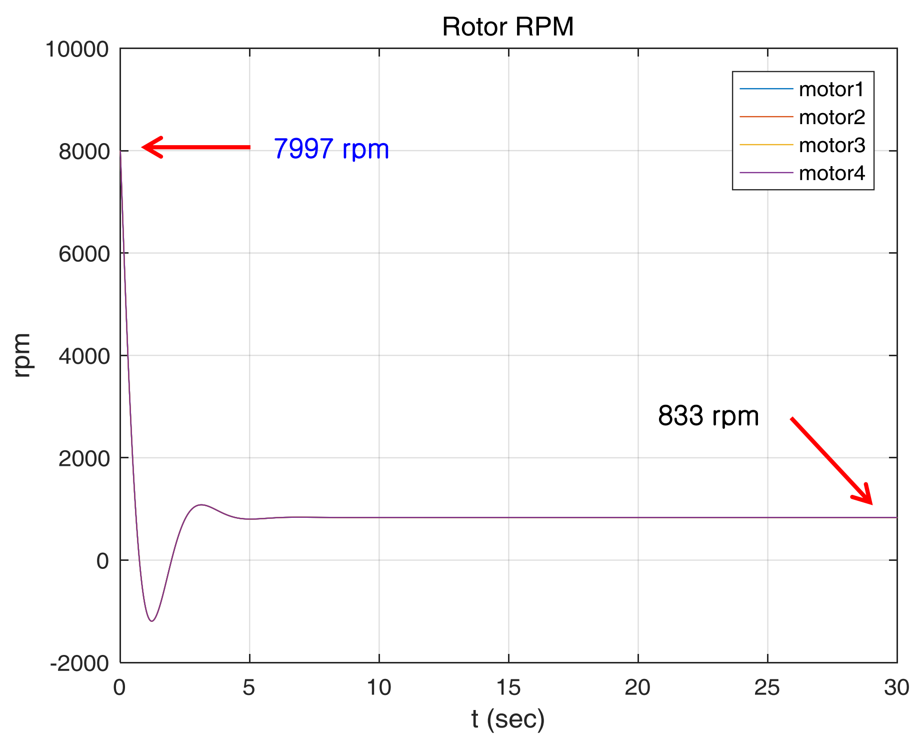
## 2. Weighted Least Square Method.

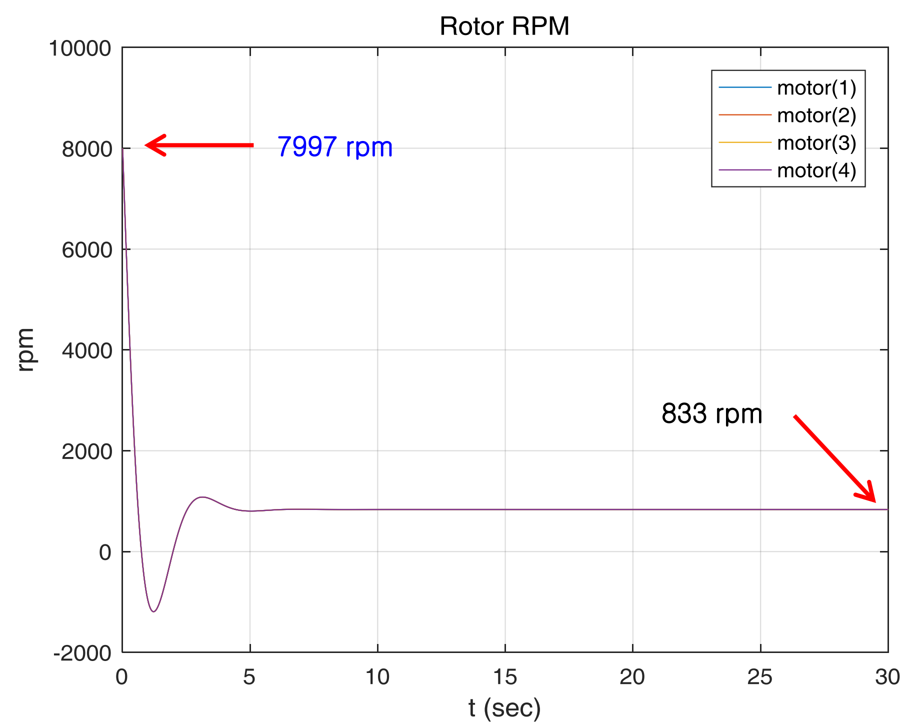
1) Quarotor + 4_CMGs Simulation.

$$J : \underbrace{w_1}_{100} \overbrace{\|a\|^2}^{\text{Motor}} + \underbrace{w_2}_{0.01} \overbrace{\|b\|^2}^{\text{Gimbal}}$$

$$x_{LSM} = A^{-1}(AA^T)^{-1}y$$

$$x_{WLSM} = W^{-1}A^{-1}(AW^{-1}A^T)^{-1}y$$



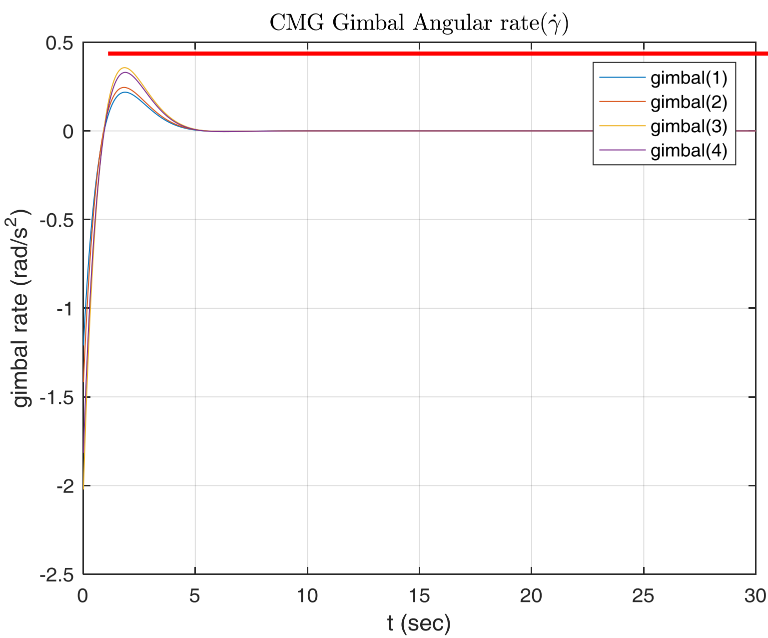



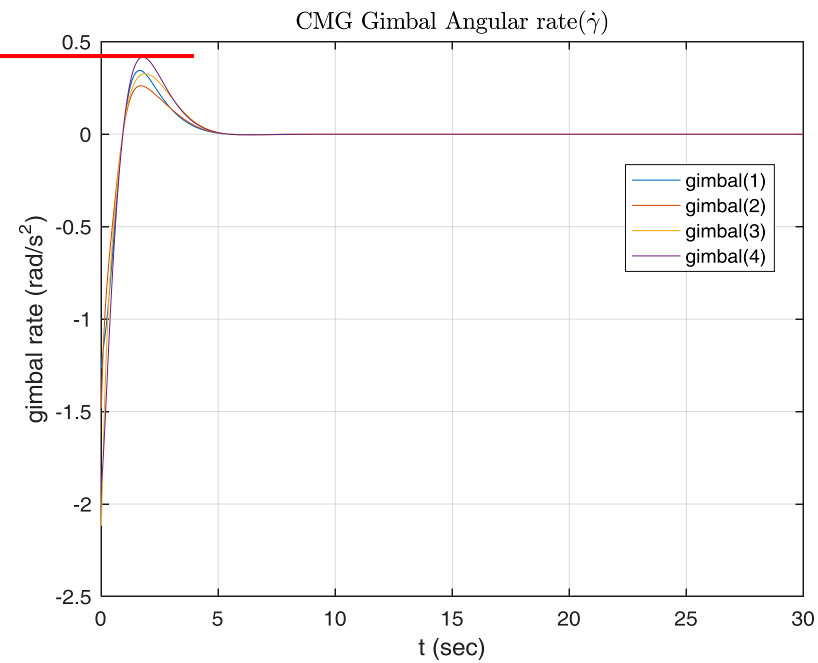
# 3. 구동기의 최적 작동 법칙.

## 2. Weighted Least Square Method.

1) Quarotor + 4_CMGs Simulation.

$$J : \underset{\uparrow \atop 100}{w_1} \left\| \overset{\text{Motor} \atop \downarrow}{a} \right\|^2 + \underset{\uparrow \atop 0.01}{w_2} \left\| \overset{\text{Gimbal} \atop \downarrow}{b} \right\|^2$$




| | | | | |
|---|---|---|---|---|
| t = 0 | -1.2098 | -1.4171 | -2.0223 | -1.8150 |
| t = 30 | 1.9407e-13 | 2.1533e-13 | 3.1665e-13 | 2.9540e-13 |

| | | | | |
|---|---|---|---|---|
| t = 0 | -1.2667 | -1.4839 | -2.1180 | -1.9008 |
| t = 30 | 2.0036e-13 | 2.3485e-13 | 3.3550e-13 | 3.0101e-13 |

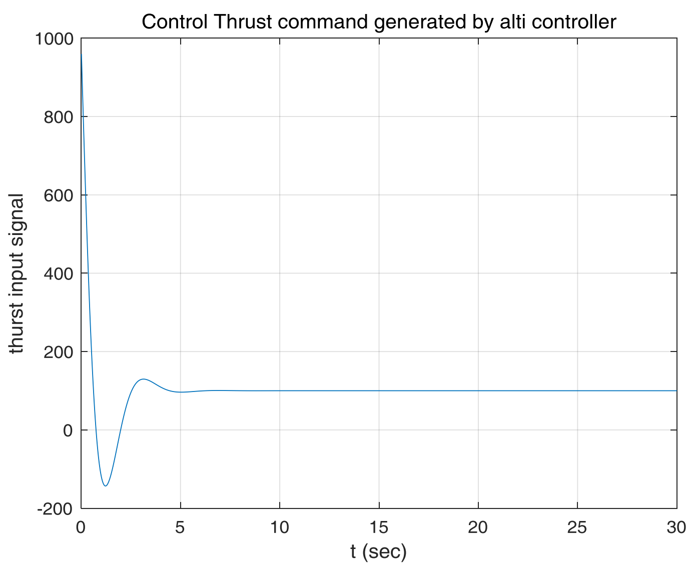
# 3. 구동기의 최적 작동 법칙.

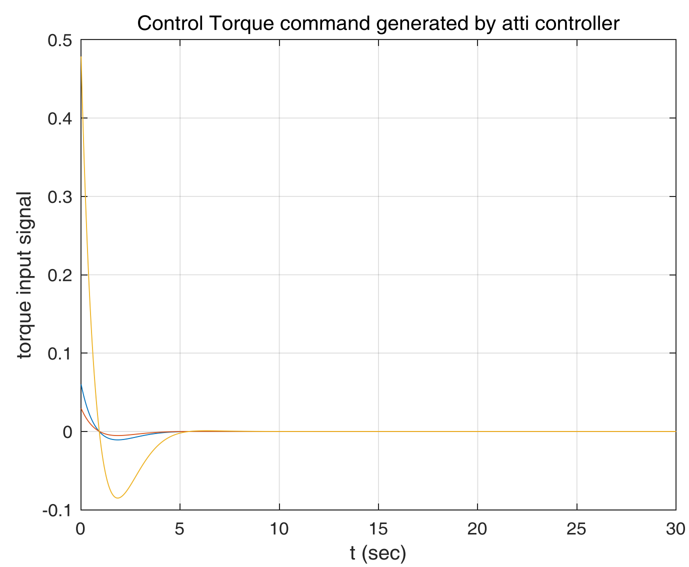
## 2. Weighted Least Square Method.

1) Quarotor + 4_CMGs Simulation.

$$J : \underset{100}{w_1}\,\|\overset{\text{Motor}}{a}\|^2 + \underset{0.01}{w_2}\,\|\overset{\text{Gimbal}}{b}\|^2$$

$$Ax = y \ : \ \begin{bmatrix} 0 & -Lk_t & 0 & Lk_t & c\beta & s & c\beta & s \\ Lk_t & 0 & -Lk_t & 0 & s & c\beta & s & c\beta \\ -d & d & -d & d & s\beta & s\beta & s\beta & s\beta \\ k_t & k_t & k_t & k_t & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ \dot{\gamma}_3 \\ \dot{\gamma}_4 \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \mathrm{T} \end{bmatrix}$$



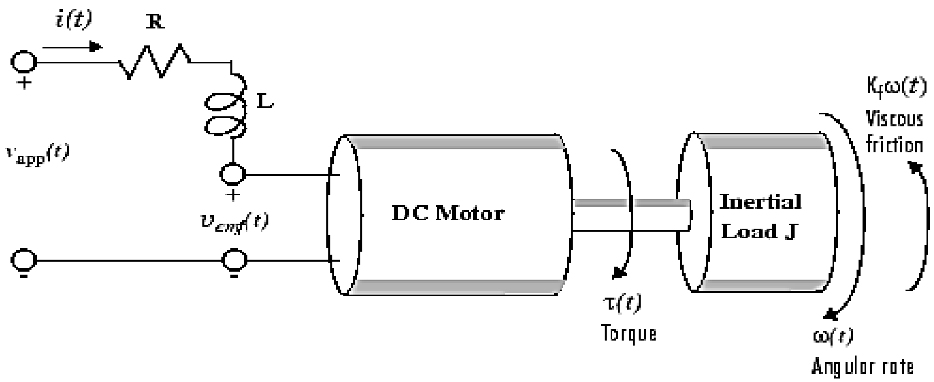



# Contents

# 4. 쿼드로터의 실제 모터제어.

## 1. Motor Dynamics.

A Simple Model of a DC Motor Driving an Inertial Load




$$\boxed{\tau(t)_m = K_m * i(t)}$$

Motor's
Mechanical Dynamics

$$J_{Tm}\dot{\omega}_m = T_m - \cancel{T_L}$$

$T_m$ = motor torque.($K_m * i_m$)
$T_L$ = motor load.($K_f * \omega_m$)

Motor's
Electronical Dynamics

$$\underbrace{v_{app}}_{\text{전압(using PWM)}}(t) = R_m i(t) + \cancel{L_m \dot{i}(t)} + \underbrace{K_b \omega_m}_{v_{emf}}$$

$R_m$ = motor resistance.
$L_m$ = motor inductance.
$K_b$ = back EMF constant.
$K_m$ = torque constant..
$K_f$ = friction coefficient.

# 4. 쿼드로터의 실제 모터제어.

## 1. Motor Dynamics.

Motor's
Electronical Dynamics

$$i(t) = \frac{1}{R_m}(-K_b\omega_m + v_{app})$$

Motor's
Mechanical Dynamics

$$J_{Tm}\dot{\omega}_m = K_m * i(t)$$

$R_m$ = motor resistance.
$L_m$ = motor inductance.
$K_b$ = back EMF constant.
$K_m$ = torque constant..
$K_f$ = friction coefficient.

Motor Dynamics

$$\dot{\omega}_m = \frac{K_m}{J_{Tm}R_m}(-K_b\omega_m + v_{app})$$

**Control input**

# 4. 쿼드로터의 실제 모터제어.

## 2. Final Quadrotor Dynamics.

Control Input

$$\mathbf{u} = -\mathbf{K}_p \mathbf{q}_e - \mathbf{K}_d \boldsymbol{\omega}$$

Ax = y :

$$\begin{bmatrix} 0 & -Lk_t & 0 & Lk_t & c\beta & s & c\beta & s \\ Lk_t & 0 & -Lk_t & 0 & s & c\beta & s & c\beta \\ -d & d & -d & d & s\beta & s\beta & s\beta & s\beta \\ k_t & k_t & k_t & k_t & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ \dot{\gamma}_3 \\ \dot{\gamma}_4 \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \mathrm{T} \end{bmatrix}$$

Actuator Input

$$\mathbf{v} = \mathbf{W}^{-1}\mathbf{A}^{-1}(\mathbf{A}\mathbf{W}^{-1}\mathbf{A}^{-1})^{-1}\mathbf{u}$$

Motor Motion

$$\dot{\omega}_m = (J_{Tm} R_m)^{-1} * K_m(-K_b \omega_m + v_{app})$$

$$\tau_\phi = U_1 = Lk_t(\omega_4^2 - \omega_2^2)$$

$$\tau_\phi = U_2 = Lk_t(\omega_1^2 - \omega_3^2)$$

$$\tau_\psi = U_3 = d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2)$$

$$T_B = U_4 = k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

Linear Motion

$$\mathbf{V}_I = R_B^I \mathbf{V}_B$$

$$\dot{\mathbf{V}}_B = (R_I^B \mathbf{F}_{G,I} + \mathbf{T}_{T,B} + \mathbf{F}_{A,B} + \Delta\mathbf{F}_{D,B}) / m - (\boldsymbol{\Omega}_B \times \mathbf{V}_B)$$

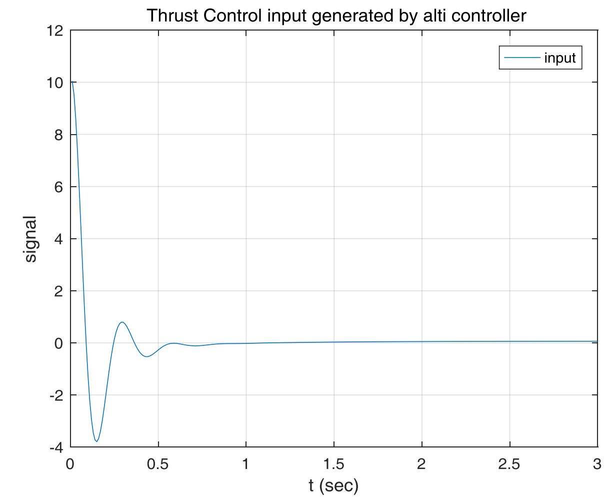
Rotational Motion

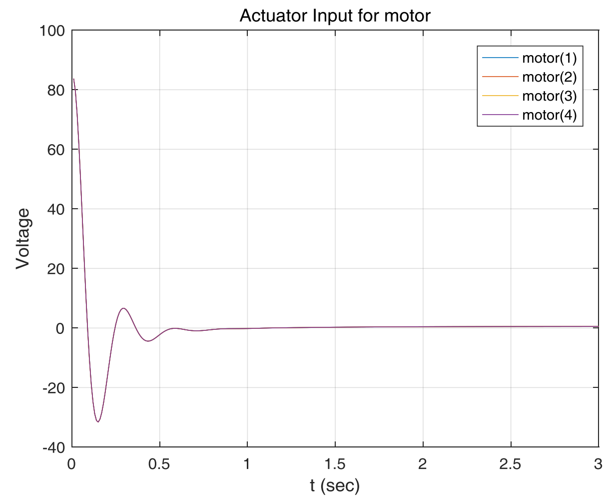
$$\boldsymbol{\Omega}_I = N_B^I \boldsymbol{\Omega}_B$$

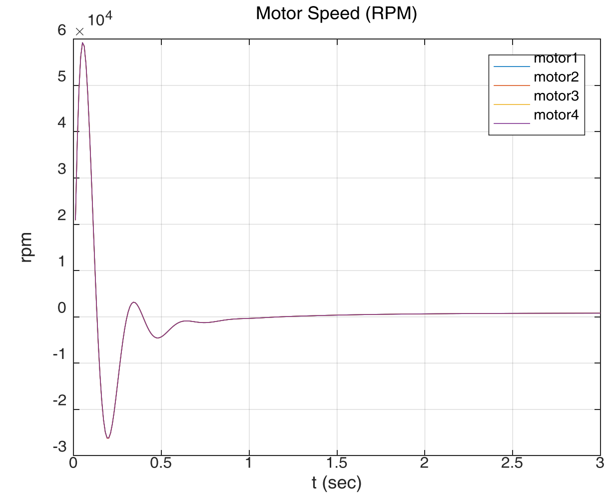
$$\dot{\boldsymbol{\Omega}}_B = J^{-1}\{\boldsymbol{\tau}_{\tau,B} - (\boldsymbol{\Omega}_B \times J\boldsymbol{\Omega}_B) - \boldsymbol{\Omega}_B^{\times}\mathbf{C}_\omega(\gamma)\mathbf{h}_\omega(\bar{\varpi}) - \mathbf{C}_\gamma(\gamma)\mathbf{H}_\omega(\bar{\varpi})\dot{\boldsymbol{\gamma}}\}$$

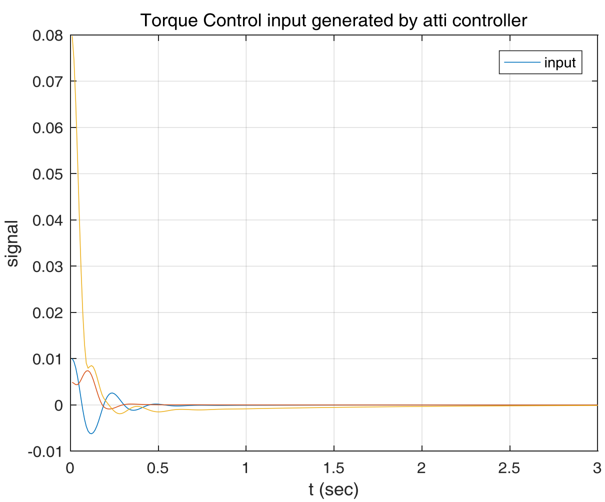
# 4. 쿼드로터의 실제 모터제어.

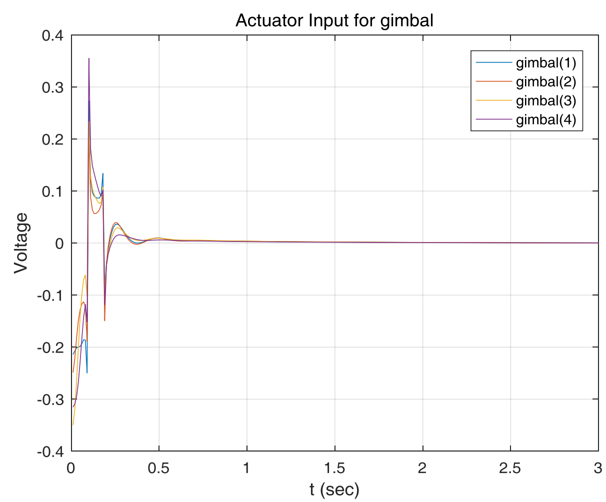
## 3. Simulation.

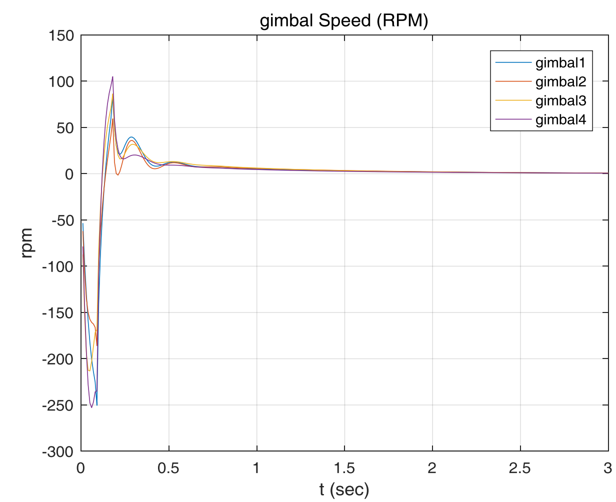


# 5. Future Plan.

1. **Motor Constrain.**
2. **Input Time delay.**
3. **Position Control.**
4. **Singularity avoidance.**
5. **Real Test.**

# Thanks for your attention!