

## Contents

---

- [Parameters](#)
- [Numerical Solution](#)
- [Closed-Form Solution](#)
- [Plot](#)
- [Functions](#)

```
clear all; clc; close all;  
% hold on;
```

## Parameters

---

```
% Set frame of view (time)  
t_end = .25; % s  
  
% Inertia Value of motor  
J_m = 5*10^(-7); % kgm^2  
  
% Stall Torque  
t_s = 0.17 * 9.81 / 100; % Nm  
% No load speed  
w_nl = 8200 * (2*pi) / 60; % rads / s
```

## Numerical Solution

---

```
% Set frame of view (time)  
tspan = [0, t_end];  
y0 = [0, 0];  
  
% Numerical solution definition  
[t, y] = ode45(@(t,y) odefcn(t, y, J_m, t_s, w_nl), tspan, y0);  
  
% Set correct units  
ts = t(:) .* 1000; % ms  
y(:,2) = y(:,2) ./ (2*pi) * 60; % rpm  
  
% plot(ts, y(:,2))
```

## Closed-Form Solution

---

```
% Set frame of view (time)  
t = linspace(0, t_end, t_end*100);  
  
% Closed-form Solution  
w = velocityProfile(J_m, t_s, w_nl); % Change inertia here  
  
% Set correct units  
ts = t(:) .* 1000;  
  
% plot(ts, w(t))
```

## Plot

---

```
% title("Numerical Solution vs Closed-Form Solution")
% xlabel("Time (ms)");
% ylabel("RPM");
% legend("Numerical Sol", "Closed-form", "Location", "southeast");
```

---

## Functions

---

```
function dydt = odefcn(t, y, J, t_s, w_n1)
    % Return numerical solution to be used with ODE45
    dydt = zeros(2,1);
    dydt(1) = y(2);
    dydt(2) = (t_s / J) * (1 - y(2)/w_n1);
end

function w = velocityProfile(J, t_s, w_n1)
    % Return velocity profile of motor with J inertia, t_s stall torque,
    % and w_n1 no load speed
    tr = w_n1 * J / t_s;
    w = @(t) w_n1 * (1 - exp(-t/tr)) / (2*pi) * 60;
end
```

---