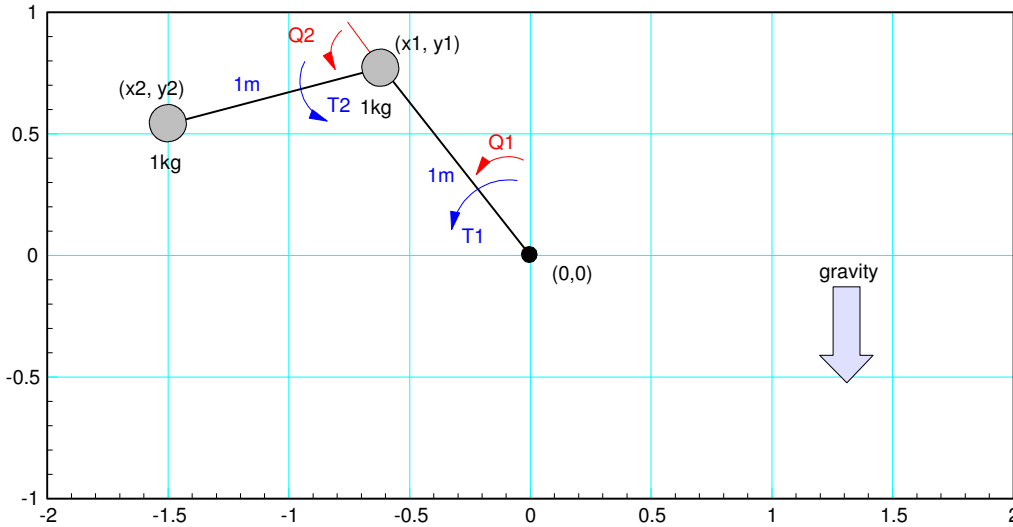# Position Control of a Robot with Gears

Consider the problem of controlling the tip-position of a 2-link robotic arm.

From before, the dynamics of the robotic arm are:

$$
\begin{bmatrix} 2(1+c_2) & (1+c_2) \\ (1+c_2) & 1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} 2s_2\dot{\theta}_1\dot{\theta}_2 + s_2\dot{\theta}_2^2 \\ -s_2\dot{\theta}_1^2 \end{bmatrix} - g \begin{bmatrix} 2c_1 + c_{12} \\ c_{12} \end{bmatrix}
$$

Assume each link is driven by a DC servo motor through a gear with a 300:1 turn reduction. Relative to the DC motor, the robot dynamics look like a disturbance, reduced by a factor of 300:

$$
\begin{bmatrix} T_{1d} \\ T_{2d} \end{bmatrix} = \frac{1}{300}\left(- \begin{bmatrix} 2(1+c_2) & (1+c_2) \\ (1+c_2) & 1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 2s_2\dot{\theta}_1\dot{\theta}_2 + s_2\dot{\theta}_2^2 \\ -s_2\dot{\theta}_1^2 \end{bmatrix} - g \begin{bmatrix} 2c_1 + c_{12} \\ c_{12} \end{bmatrix} \right)
$$

With this reduction of 300x, the problem of controlling the motion of a robotic arm becomes simply the problem of controlling a DC motor with a slight disturbance.

Equations for a Brush-Type DC Motor (DC Servo Motor)

The two are coupled by the torque constant, Kr:
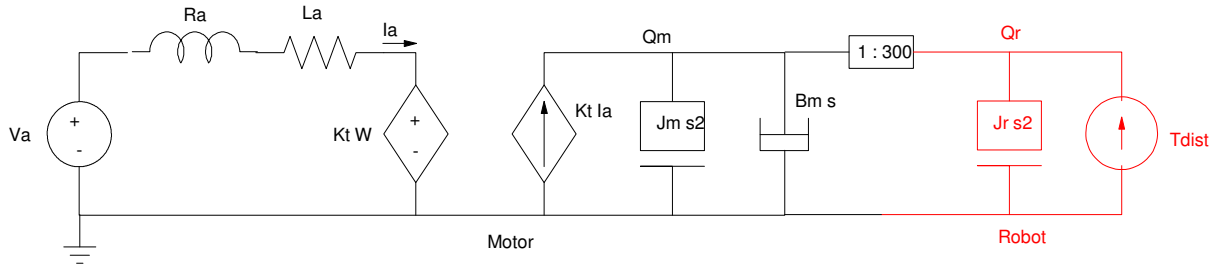
Model for a DC Servo Motor connected to a robot with a 300:1 gear reduction

From before, the dynamics for a DC servo motor are:

$$\left(\left(J_m + \frac{J_r}{300^2} \right) s^2 + B_m s \right) \theta_m = K_t I_a + \frac{T_{dist}}{300}$$

$$V_a = (Ls + R)I_a + K_t s \theta_m$$

which simplifies to

$$\theta = \left(\frac{1}{s} \right) \left(\frac{K_t}{(Js+D)(L_a s + R_a) + K_t^2} \right) V_a + \left(\frac{Ls+R}{300s} \right) T_{dist}$$

or, assuming L = 0

$$\theta = \left(\frac{1}{s} \right) \left(\frac{K_t}{(Js+D)(L_a s + R_a) + K_t^2} \right) V_a + \left(\frac{R}{300s} \right) T_{dist}$$

For illustration purposes, assume the DC motor is a Pittman 14207 Motor operated at 76.4DC. (100 Watts)

Specification	Units	Part/Model Number
		14207 15.2 V	14207 19.1 V	14207 24.0 V	14207 30.3 V	14207 38.2 V	14207 48.0 V	14207 60.0 V	14207 76.4 V
Supply Voltage	VDC	15.2	19.1	24.0	30.3	38.2	48.0	60.6	76.4
Continuous Torque	oz-in	50	50	50	50	50	50	50	50
	Nm	0.353	0.353	0.353	0.353	0.353	0.353	0.353	0.353
Speed @ Cont. Torque	RPM	2910	2770	2810	2810	2820	2840	2830	2830
Current @ Cont. Torque	Amps (A)	9.98	7.48	5.98	4.71	3.74	2.99	2.37	1.87
Continuous Output Power	Watts (W)	108	102	104	104	104	105	105	105
Motor Constant	oz-in/sqrt W	12	13	13	13	13	13	13	13
	Nm/sqrt W	0.085	0.092	0.092	0.092	0.092	0.092	0.092	0.092
Torque Constant	oz-in/A	6.00	8.004	10.0	12.7	16.0	20.0	25.3	32.0
	Nm/A	0.042	0.057	0.071	0.09	0.113	0.141	0.179	0.226
Voltage Constant	V/krpm	4.44	5.92	7.39	9.39	11.8	14.8	18.7	23.7
	V/rad/s	0.042	0.057	0.071	0.09	0.113	0.141	0.179	0.226
Terminal Resistance	Ohms	0.24	0.39	0.59	0.93	1.46	2.29	3.64	5.78
Inductance	mH	0.31	0.56	0.87	1.40	2.23	3.48	5.58	8.93
No-Load Current	Amps (A)	0.51	0.38	0.30	0.24	0.19	0.15	0.12	0.090
No-Load Speed	RPM	3330	3140	3160	3140	3140	3160	3150	3140
Peak Current	Amps (A)	63.3	49.0	40.7	32.6	26.2	21.0	16.6	13.2
Peak Torque	oz-in	377	389	404	411	416	416	418	420
	Nm	2.6616	2.7463	2.8522	2.9017	2.937	2.937	2.9511	2.9652
Coulomb Friction Torque	oz-in	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2
	Nm	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155	0.0155
Viscous Damping Factor	oz-in/krpm	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
	Nm s/rad	1.68E-5	1.68E-5	1.68E-5	1.68E-5	1.68E-5	1.68E-5	1.68E-5	1.68E-5
Electrical Time Constant	ms	1.3	1.4	1.5	1.5	1.5	1.5	1.5	1.5
Mechanical Time Constant	ms	6.3	5.8	5.6	5.5	5.4	5.4	5.4	5.4
Thermal Time Constant	min	32	32	32	32	32	32	32	32
Thermal Resistance	Celsius/W	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
Max. Winding Temperature	Celsius	155	155	155	155	155	155	155	155
Rotor Inertia	oz-in-sec2	0.0067	0.0067	0.0067	0.0067	0.0067	0.0067	0.0067	0.0067
	kg-m2	4.73E-5	4.73E-5	4.73E-5	4.73E-5	4.73E-5	4.73E-5	4.73E-5	4.73E-5
Weight (Mass)	oz	54.5	54.5	54.5	54.5	54.5	54.5	54.5	54.5
	g	1545	1545	1545	1545	1545	1545	1545	1545

http://www.pittman-motors.com/Brush-DC-Motors.aspx

This motor has a maximum torque of 0.353Nm. With a 300:1 gear reduction, it can output 105Nm. The maximum load for the robotic arm is when the arms are outstretched, at which point the torque is

$$T = 1kg * 1m + 1kg * 2m = 3Nm$$

So, this motor along with the gear is capable of holding the robotic arm straight out with a safety margin of 35.1. The motor looks like a reasonable choice.

From the table, the dynamics are:

- Kt = 0.226 Nm/A
- L = 8.93mH
- J = 4.73E-5 kg m2
- R = 5.78 Ohms
- D = unknown
- No-Load Speed = 3140 rpm @ 0.090 A

You can compute the resistance through a power balance

$$P_{in} = 76.4V \cdot 90mA = 6.88W$$

The electrical losses are

$$P_R = I_a^2 R = (90mA)^2 \cdot 5.78\Omega = 46.8mW$$

The remaining losses must be the rotational losses

$$T \cdot \omega = 6.829W$$

The no-load speed is 3140 rpm (328.8 rad/sec)

The torque is

$$T = \frac{6.829W}{328.8 rad/\sec} = 0.02077Nm$$

This then tells you the friction

$$T = D\omega$$

$$D = \frac{T}{\omega} = 63.17E - 6 \; \frac{\text{Nm}}{\text{rad/sec}}$$

Plugging in numbers:

$$\omega = \left(\frac{K_t}{(Js+D)(L_a s + R_a) + K_t^2} \right) V_a$$

$$\omega = \left(\frac{535,050}{s^2 + 648.6s + 121,790} \right) V_a$$

The step response of a motor with no load is then

```
>> G = tf(535050, [1, 648.6, 121790])


       535050
----------------------
s^2 + 648.6 s + 121790
```

The 2% settling time is approximately 4 / the real part of the dominant pole

```
>> Ts = 4 / 324

   0.0123
```

Plot the step response for a 50V step out to 20ms

```
>> t = [0:0.001:1]' * 0.02;
>> y = step(G,t);
>> plot(t*1000,y * 50);
>> xlabel('Time (ms)');
>> ylabel('Speed (rad/sec)')
```
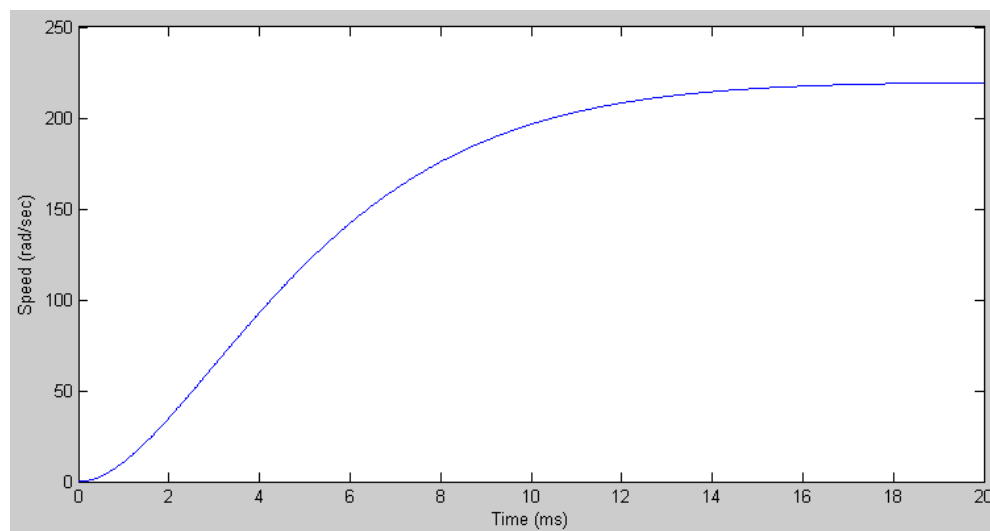
Step Response of a Pittman 14207 Motor with No Load and a 50V Step Input

Motors and Gears:

One problem with using a motor to drive a robotic arm is that motors are designed to spin whereas robots tend to remain at a certain spot. To couple a motor to a robot, a gear is often used.

Gear Reducer: http://motiontek.ca/gear_reducer.html

Gears reduce the speed of the shaft (or conversely, a slow speed at the output corresponds to a high speed at the input - which is good since motors want to spin.) The relationship for a gear is

$$N_1 \omega_1 = N_2 \omega_2$$

or

$$\omega_1 = \left(\frac{N_2}{N_1}\right)\omega_2$$

With a 12:1 gear reduction (N1 = 12, N2 = 1), the output shaft spins 12x slower than the input shaft.

Power has to balance, meaning

$$T_1\omega_1 = T_2\omega_2$$

Substituting

$$T_1 = \left(\frac{N_1}{N_2}\right)T_2$$

With a 12:1 gear reduction, the output shaft has 12x higher torque than the input shaft. Gears act as torque amplifiers.

One interesting feature of gears is they reduce the impedance's as seen by the motor. Using the analogy

$$V = IR$$

$$\omega_2 = T_2 Z_2$$

then an impedance at the output of the gear (Z2) looks like

$$\left(\frac{N_1}{N_2}\right)\omega_1 = \left(\frac{N_2}{N_1}\right)T_1 Z_2$$

$$\omega_1 = T_1\left(\left(\frac{N_2}{N_1}\right)^2 Z_2\right)$$

The impedance as seen through a gear increase as the turn-ratio squared.

The robot has an inertia between 0 and 3 kg m2. Assuming a 300:1 turn ratio along with a gear with the same parameters, the net inertia seen by the motor is

$$J_{total} = J_{motor} + J_{gear} + \frac{J_{robot}}{300^2}$$

$$J_{total} = (47.3 \cdot 10^{-6}) + (?) + \left(\frac{0..3}{300^2}\right);kg \cdot m^2$$

$$47.3 \cdot 10^{-6} < J_{total} < 80.63 \cdot 10^{-6};kg \cdot m^2$$

Assume the gear's inertia is 20% of the motor's inertia (a guess) and the robot's inertia is 1.5 kg m2. Then, the total inertia is
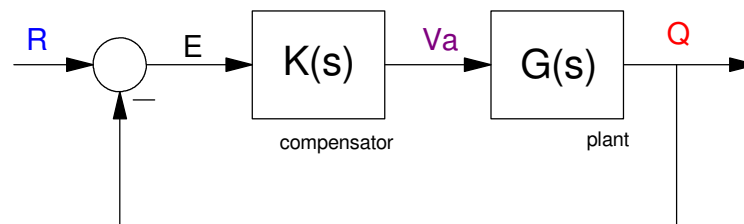
$$J_{total} \approx 73 \cdot 10^{-6};kg \cdot m^2$$

The net dynamics for each motor are then

$$\theta \approx \left(\frac{346{,}700}{s^3 + 648.1s^2 + 78{,}910s} \right) V_a + \left(\frac{5.78}{300s} \right) T_{dist}$$

$$\theta \approx \left(\frac{346{,}700}{s(s+485.6)(s+162.5)} \right) V_a + \left(\frac{0.0192}{s} \right) T_{dist}$$

Feedback Control of a DC Motor

Assume a unity feedback control law where Ref is the desired angle (as defined by the inverse kinematics from before):

where
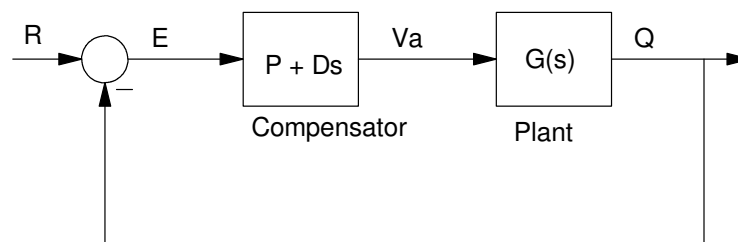
$$G(s) = \left(\frac{346{,}700}{s(s+485.6)(s+162.5)} \right)$$

A lead compensator for K(s) would speed up the system

$$K(s) = k \left(\frac{s+162.5}{s+1625} \right)$$

resulting in

$$GK = \left(\frac{346{,}700k}{s(s+485.6)(s+1625)} \right)$$

Alternatively, a PD feedback control law could be used

PD Control Law of a DC Servo Motor

resulting in the open-loop system being

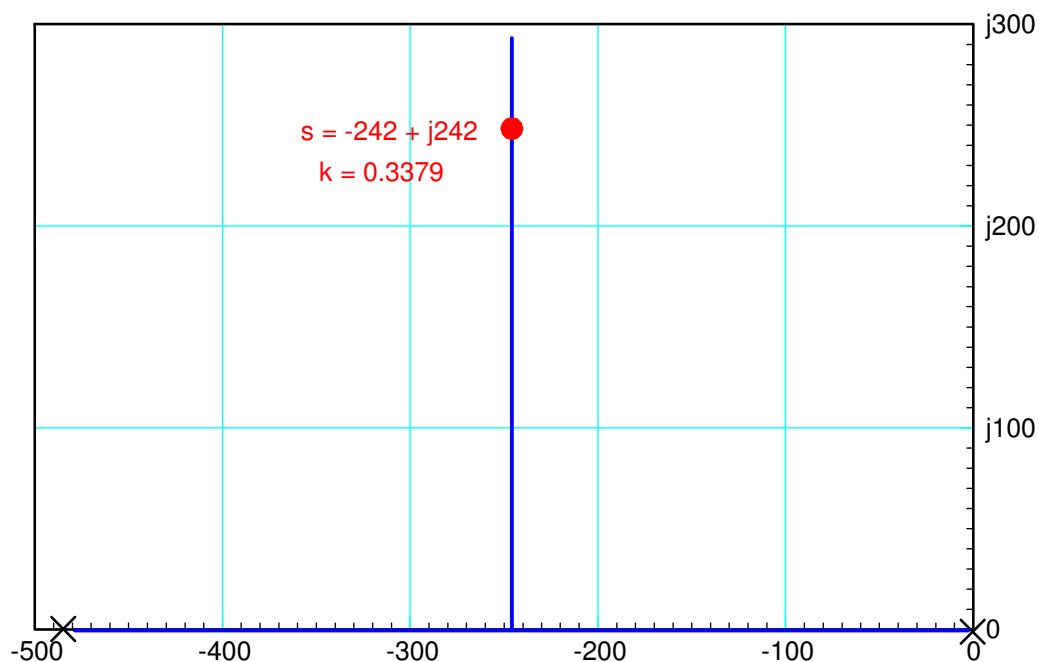$$GK = \left(\frac{346,700(P+Ds)}{s(s+485.6)(s+1625)} \right)$$

Selecting

$$(P+Ds) = k(s+162.5)$$

results in the root locus following the roots of open-loop system being

$$s(s+485.6)+k = 0$$

with the following root locus plot:

Root Locus of a DC Servo Motor with (P + Ds) = k (s + 162)

To place the closed-loop poles at s = -242 + j242 (meaning 4% overshoot for a step input or a damping ratio of 0.7),

$$\left(\frac{346,700k}{s(s+485.6)} \right)_{s=-242+j242} = -1$$

$$k = 0.3379$$

so the PD controller is

$$P+Ds = 0.3379(s+162.5)$$

$$P+Ds = 0.3379s + 54.91$$

D = 0.3379

P = 54.91

The step response for the closed-loop system is then the step response of

$$\theta = \left(\frac{117{,}128}{(s+242+j242)(s+242-j242)} \right) R$$

To evaluate the effect of the disturbance, evaluate the following

$$\theta \approx \left(\frac{346{,}700}{s(s+485.6)(s+162.5)} \right) V_a + \left(\frac{0.0192}{s} \right) T_{dist}$$

With

$$V_a = (P + Ds)(R - \theta)$$

at DC (s=0) you get

$$s\theta = 0 = \left(\frac{346{,}700}{(485.6)(162.5)} \right) (P)(R - \theta) + (0.0192)T_{dist}$$

When R = 0, the disturbance effects the angle as

$$\theta = \left(\frac{0.0192}{4.394P} \right) T_{dist}$$

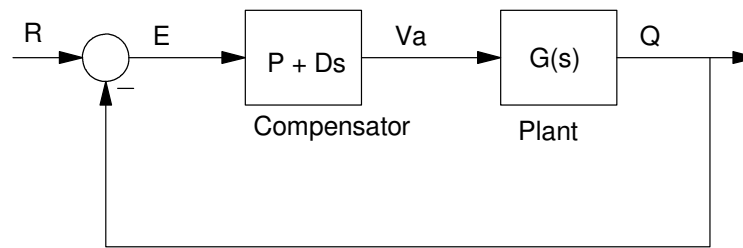Worst case when the disturbance is 3Nm (arm pointing straight out) and P = 54.91

$$\theta = 79.58 \cdot 10^{-6} \text{ rad} = 0.0045^0$$

With this PD controller, the disturbance will only offset the angle by 0.0045 degrees (worst case).

With a 300:1 gear reduction, all you need to do is control the angle of the motor.

Motor Control Case 1: PD Control

Assume a PD controller of the following form

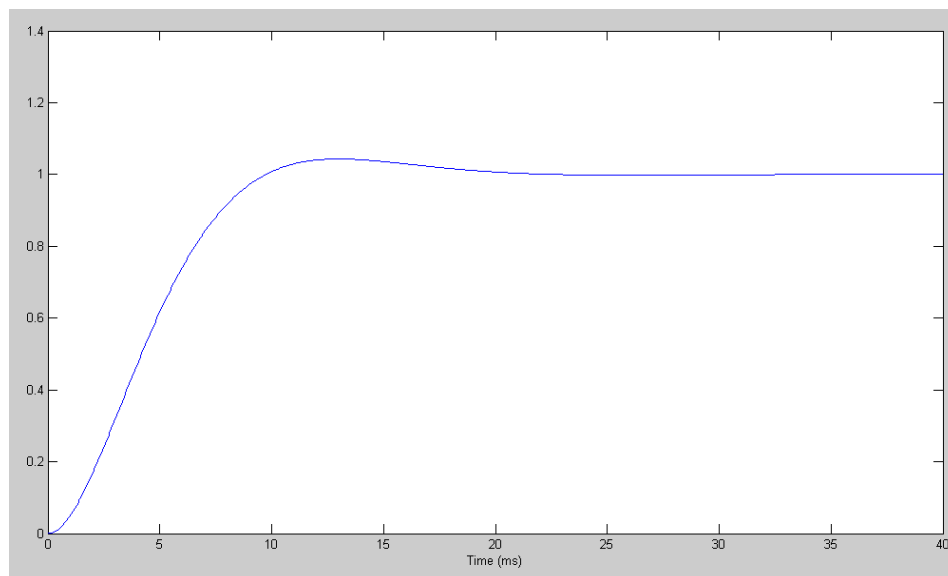PD Control with Va = (P + Ds) (R - Q)

where

$$P + Ds = 0.3379(s + 162.5)$$

The closed-loop system is then

$$\theta = \left(\frac{117{,}128}{(s+242+j242)(s+242-j242)} \right) R$$

which has the following step response:

```
>> G = zpk([],[-242+j*242,-242-j*242],117128);
>> t = [0:0.001:1]' * 0.04;
>> y = step(G,t);
>> plot(t*1000,y)
>> xlabel('Time (ms)');
```


Note that actual motor angle lags behind the desired angle (R) by 10ms. Also note that the control law is

$$V_a = (P + Ds)(R - \theta)$$

meaning that you need to input

> The desired angle (R), and

> The derivative of the desired angle (sR)

To simulate this type of controller, add dynamics so that the desired motor angle (R as computed from the inverse dynamics) differs from the acutal motor angle (θ) by the above transfer function.

PD Control (case 2):

A more common type of PD control does not use the derivative of the desired angle and looks like the following:
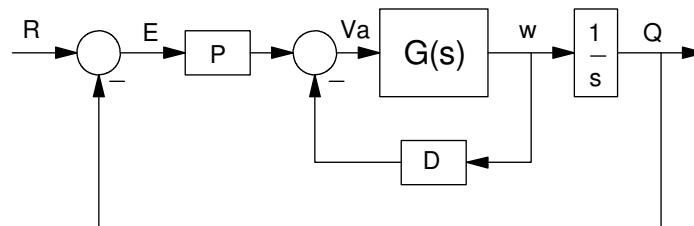
PD Control with Va = P*R - (Ds + P) Q

Here, the relationship between θ and R is

$$\theta = \left(\frac{346,700P}{s(s+485.6)(s+162.5)+346,700(Ds+P)} \right) R$$
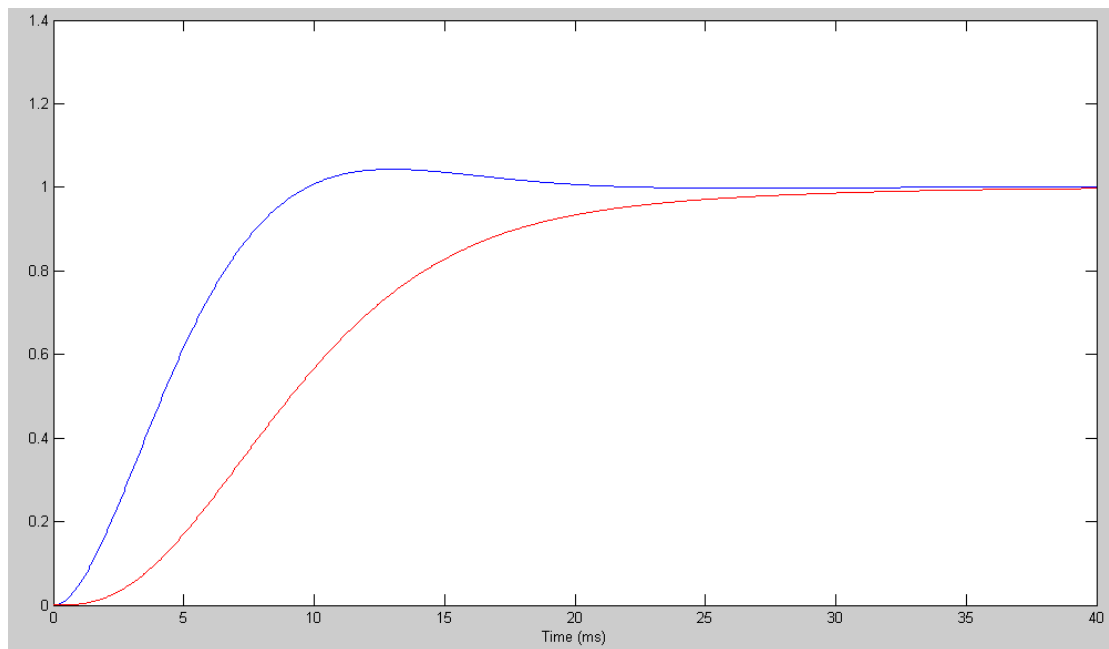
With

$$P + Ds = 0.3379(s + 162.5)$$

then

$$\theta = \left(\frac{117128}{(s+242+j242)(s+242-j242)} \right) \left(\frac{162.5}{s+162.5} \right) R$$

with the following step response

```
>> G2 = zpk([],[-242+j*242,-242-j*242, -162.5],117128 * 162.5);
>> y2 = step(G2,t);
>> plot(t*1000,y,'b',t*1000,y2,'r')
>> xlabel('Time (ms)');
```

Step Response where you use sR (blue) and do not use sR (red)

Note that
- You no longer need to include the derivative of R (good), but
- You get a longer delay between the desired angle (R) and the actual angle

PD Control plus Feed Forward Control

Another way to write the dynamics where

$$V_a = PR - (P + Ds)\theta$$

is

$$\theta = \left(\frac{117128}{(s+242+j242)(s+242-j242)} \right) \left(\frac{162.5}{s+162.5} \right) R$$

or rewriting it:

$$\theta = \left(\frac{1}{5.2539e{-}008s^3 + 3.3967e{-}005s^2 + 0.0103s + 1} \right) R$$

If you add a feedforward term so that

$$V_a = (1)(PR)$$

becomes

$$V_a = (1 + 0.0103s)(PR)$$

then the transfer function becomes

$$\theta = \left(\frac{0.0103s+1}{5.2539e{-}008s^3 + 3.3967e{-}005s^2 + 0.0103s + 1} \right) R$$

This is essentially the PD controller from before

If you add a second feedforward term
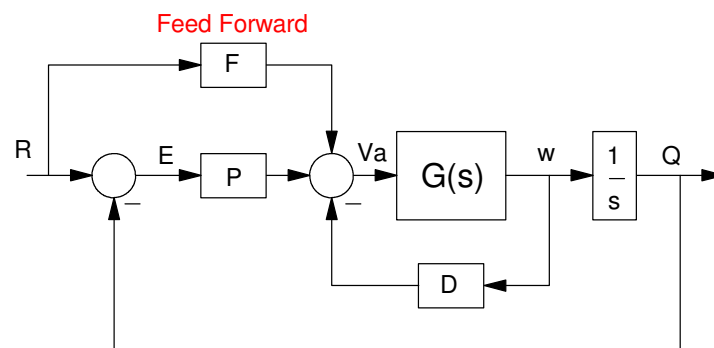
$$V_a = (1 + 0.0103s + 0.000034s^2)(PR)$$

the transfer function becomes

$$\theta = \left(\frac{3.3967e{-}005s^2 + 0.0103s + 1}{5.2539e{-}008s^3 + 3.3967e{-}005s^2 + 0.0103s + 1} \right) R$$

In this case, you need to know

- R
- sR
- s2R

The block-diagram for the feed-forward controller is

With the resulting step response:

```
>> P = poly([-242+j*242,-242-j*242,-162.5])

   1.0e+007 *

     0.0000    0.0001    0.0196    1.9033
```

```
>> format short e
>> P1 = P / P(4)

   5.2539e-008  3.3967e-005  1.0286e-002  1.0000e+000

>> G1 = tf(1, P1);
>> y1 = step(G1,t);

>> G2 = tf(P1(3:4), P1);
>> y2 = step(G2,t);

>> G3 = tf(P1(2:4), P1);
>> y3 = step(G3,t);
>> plot(t*1000,[y1,y2,y3])
>> xlabel('Time (ms)');
```
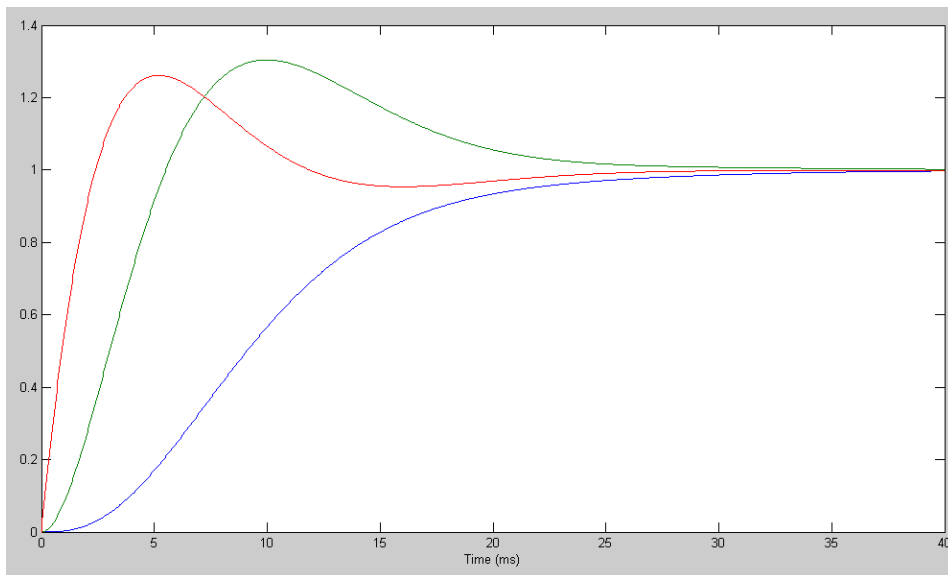
Step Response for a PD controller with the input using R (blue), R and sR (green), R, sR, s2R (red)

Note that by inputting the derivative and 2nd derivative of R, the lag between the actual angle (θ) and the commanded angle (R) gets smaller and smaller.

Simulation Results

Write a program to trace a square for a 2-link arm

```
function [P1, Qnet] = RRMove(P0, P1, T)


function [Q] = InverseRR(TIP)


function RR(Q, TIP)

% TwoLink.m
%
%  Q = motor angles
%  R = desired angles
%  TIP  = path you're trying to follow


>> P0 = [0.2; 0.2];
>> [P1, Q1] = RRMove(P0, [1; 0.2], 2);
>> [P2, Q2] = RRMove(P1, [1; 1], 2);
>> [P3, Q3] = RRMove(P2, [0.2; 1], 2);
>> [P4, Q4] = RRMove(P3, [0.2; 0.2], 2);
>> R = [Q1, Q2, Q3, Q4];
>>
>> G = zpk([],-10,10);
>> Q0 = R(:,1)

Q0 =

    2.2143
   -2.8578

>> Q = step3(G, t, Q0, R);
>> TwoLink
```
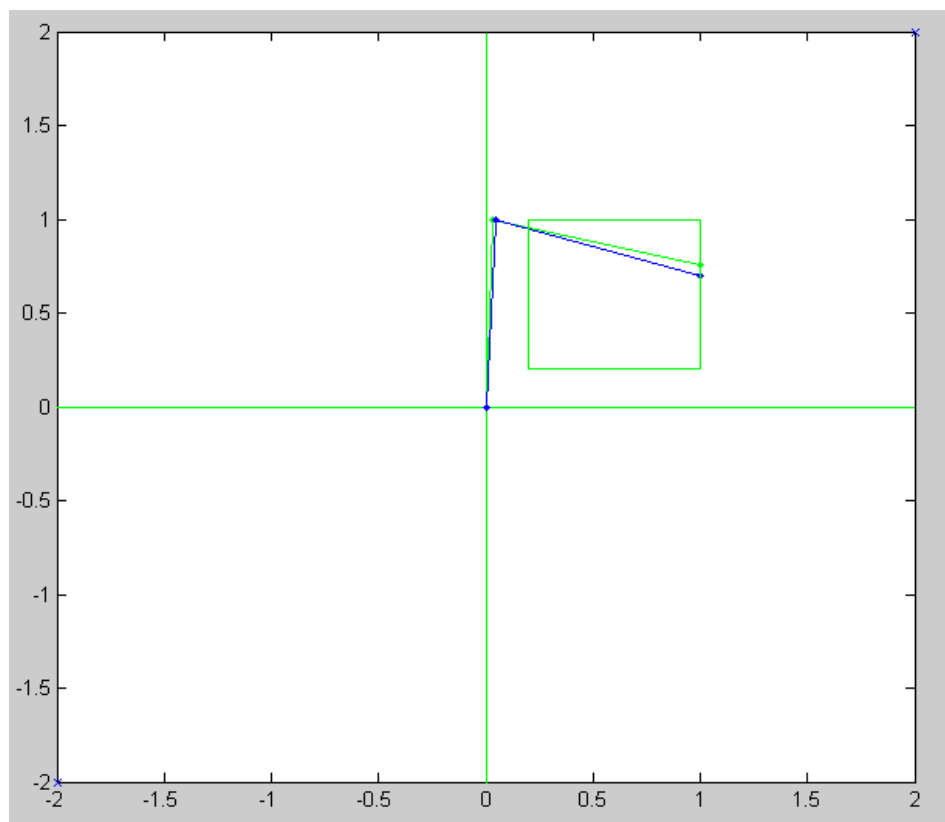
Desired Position of the Robot (green) and Actual Position (blue)