Course No. M4609301
Introduction to Robotics
機器人學概論

Matlab Chapter 9
Dynamics and Control

Shuhao Liang, Marnel Altius
Graduate Institute of Intelligent Manufacturing Technology
National Taiwan University of Science and Technology

2024

Part III Arm-Type Robots

What's the difference between kinematics and dynamics?

Kinematics is the study of how objects move (velocity, acceleration), while **dynamics** is the study of **how forces** cause objects to move:

- **Kinematics**
  - Describes the motion of an object without considering the forces that cause it. Kinematics focuses on an object's **position, velocity, acceleration**, and **time**. For example, describing the motion of an apple falling from a tree is kinematics.
- **Dynamics**
  - Studies the causes of motion by **examining the forces** that act upon an object and their effects. Dynamics is concerned with explaining why motion occurs. For example, studying a car collision or the force of gravity on a skydiver is dynamic.

Chapter

9 Dynamics and Control

- Key Points
- Independent Joint Control
- Rigid-Body Equations of Motion
- Forward Dynamics
- Rigid-Body Dynamics Compensation
- Wrapping Up

Key points

- Dynamics and Control of a serial-link manipulator arm

- The motion of the end-effector is the composition of the motion of each link
- The links are ultimately moved by forces and torques exerted by the joints.
- Each link in the serial-link manipulator is supported by a reaction **force** and **torque** from the preceding link.
- Each link is subject to its own weight as well as the reaction forces and torques from the links that it supports

Key points
- Dynamics and Control of a serial-link manipulator arm

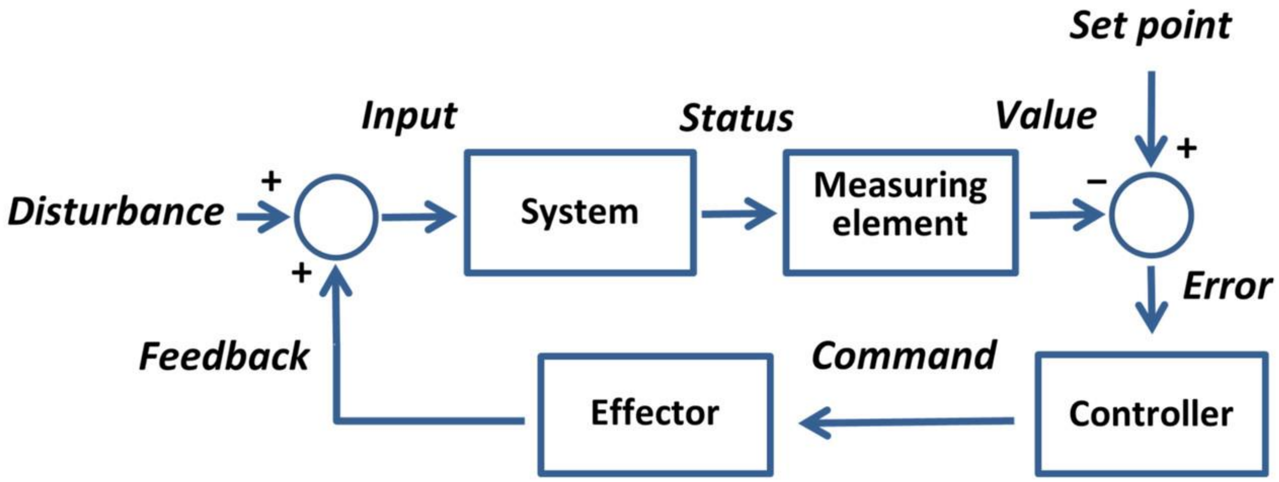
- the rigid-body equations of motion, a set of coupled dynamic equations, that describe the joint torques necessary to achieve a particular manipulator state.
- Consists of **inertia**, **gravity, load** and **gyroscopic coupling**
- The equations provide insight into how the motion of one joint exerts a disturbance force on other joints, and how inertia and gravity load varies with configuration and payload.

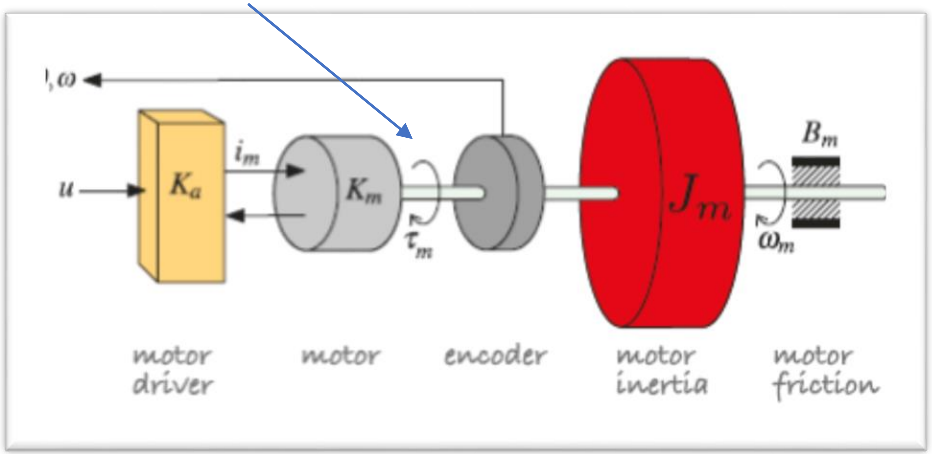
Independent Joint Control

A robot drive train comprises an actuator or motor, and a transmission to connect it to the link.

A common approach to robot joint control is to consider each joint or axis as an independent control system that attempts to accurately follow its joint angle trajectory.

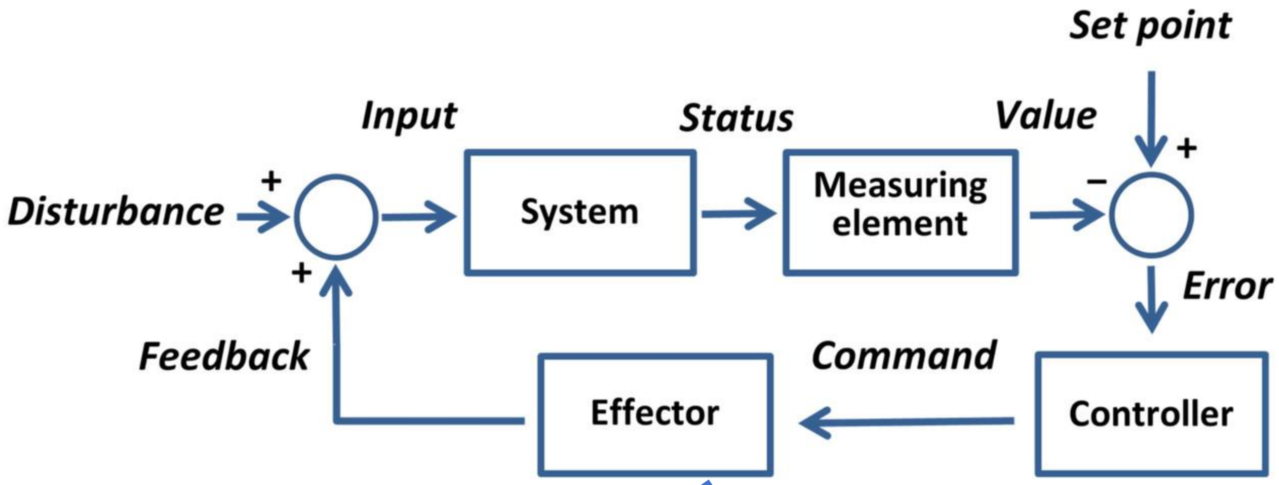
However as we shall see, this is complicated by various *disturbance* torques due to gravity, velocity and acceleration coupling, and friction that act on the joint.





Independent Joint Control

A robot drive train comprises an actuator or motor, and a transmission to connect it to the link.



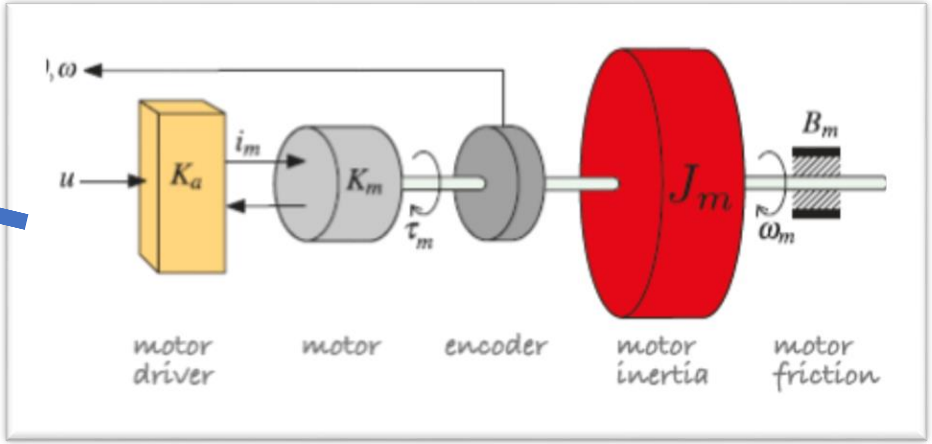
Electric motors can be either current or voltage controlled. Here we assume current control where a motor driver or amplifier provides current

$$i_m = K_a u$$

that is linearly related to the applied control voltage u and where K_a is the transconductance of the amplifier with units of $\mathrm{A\,V^{-1}}$. The torque generated by the motor is proportional to current

$$\tau_m = K_m i_m$$

where K_m is the motor torque constant with units of $\mathrm{N\,m\,A^{-1}}$. The torque accelerates the rotational inertia J_m, due to the rotating part of the motor itself, which has a rotational velocity of ω. Frictional effects are modeled by B_m.



Friction

Any rotating machinery, motor or gearbox, will be affected by friction – a force or torque that *opposes* motion. The net torque from the motor is

$$\tau' = \tau_m - \tau_f$$

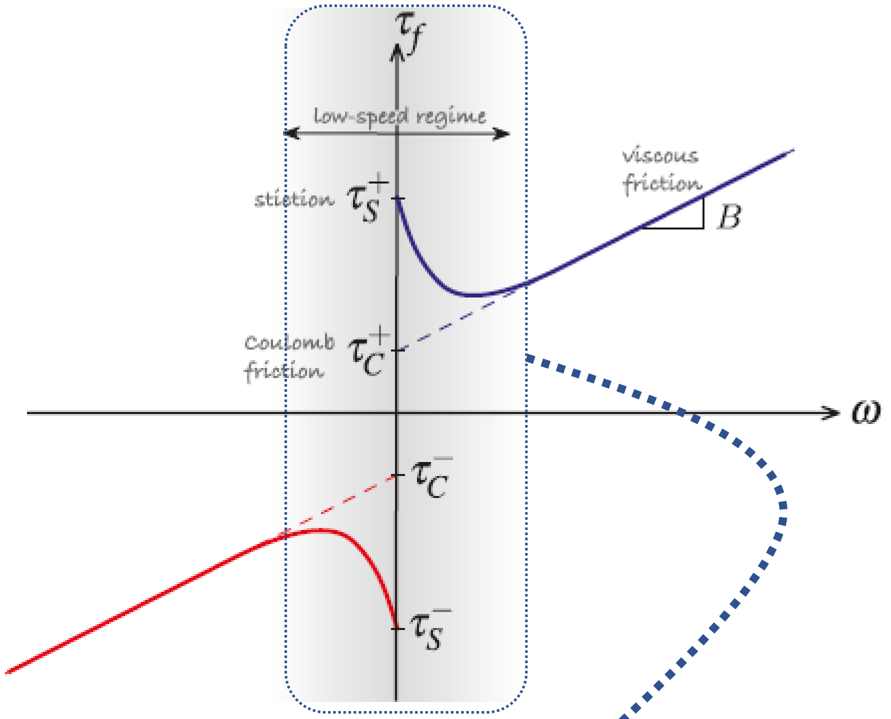
where τ_f is the friction torque which is function of velocity

$$\tau_f = B\omega + \tau_C \tag{9.1}$$

where the slope $B > 0$ is the viscous friction coefficient and the offset is Coulomb friction. The latter is frequently modeled by the nonlinear function

$$\tau_C = \begin{cases} \tau_C^+ & \omega > 0 \\ 0 & \omega = 0 \\ \tau_C^- & \omega < 0 \end{cases} \tag{9.2}$$

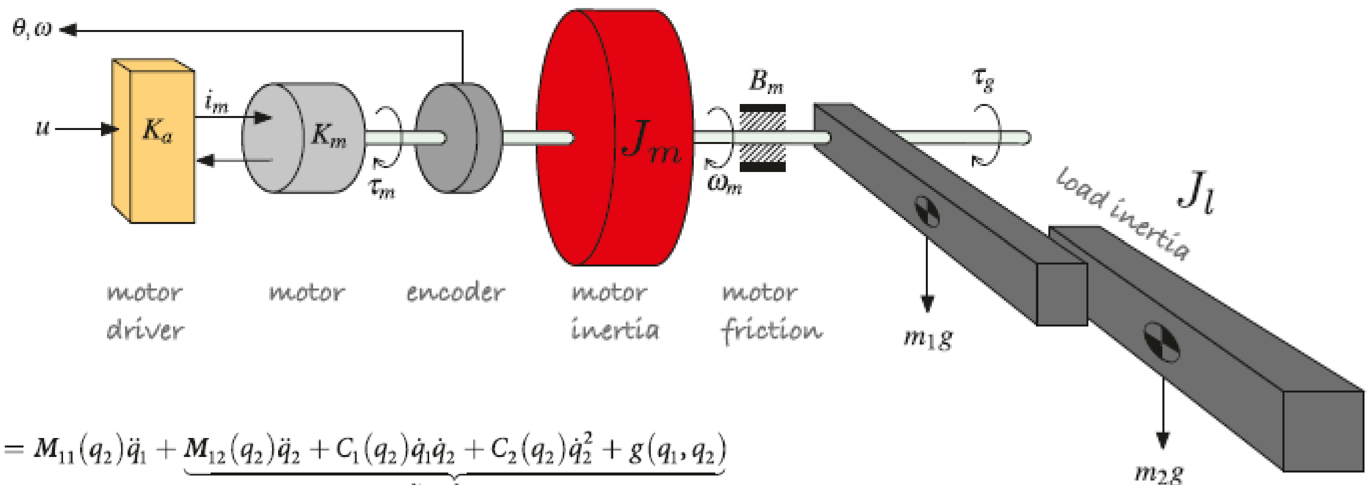
In general the friction coefficients depend on the direction of rotation and this asymmetry is more pronounced for Coulomb than for viscous friction.

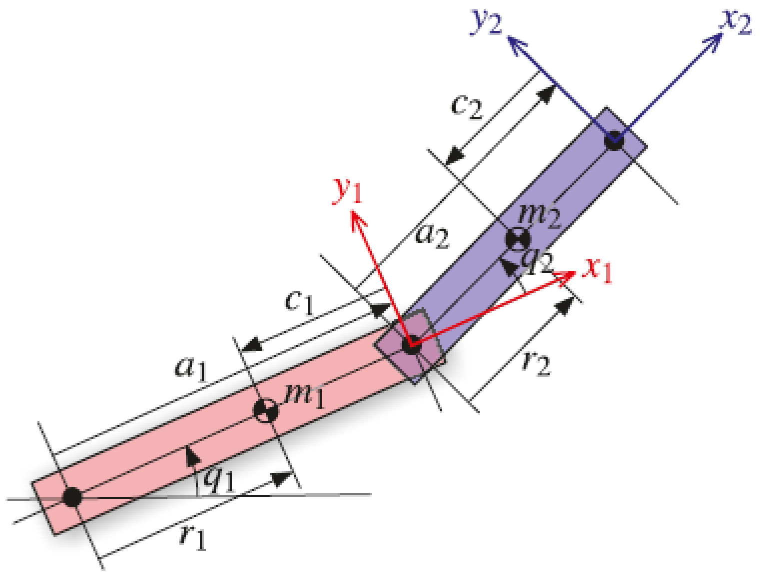


The total friction torque as a function of rotational velocity is shown in Fig. 9.2. At very low speeds, highlighted in grey, an effect known as stiction becomes evident. The applied torque must exceed the stiction torque before rotation can occur – a process known as *breaking stiction*. Once the machine is moving the stiction force rapidly decreases and viscous friction dominates.

A simple Two-joint robot

- We assumed the mass of the red link is concentrated at its center of mass (CoM)





$$\tau_1 = M_{11}(q_2)\ddot{q}_1 + \underbrace{M_{12}(q_2)\ddot{q}_2 + C_1(q_2)\dot{q}_1\dot{q}_2 + C_2(q_2)\dot{q}_2^2 + g(q_1, q_2)}_{\text{disturbance}}$$

$$M_{11} = m_1\left(a_1^2 + 2a_1c_1 + c_1^2\right) + m_2\left(a_1^2 + (a_2 + c_2)^2 + (2a_1a_2 + 2a_1c_2)\cos q_2\right)$$

$$M_{12} = m_2(a_2 + c_2)(a_2 + c_2 + a_1\cos q_2)$$

$$C_1 = -2a_1m_2(a_2 + c_2)\sin q_2$$

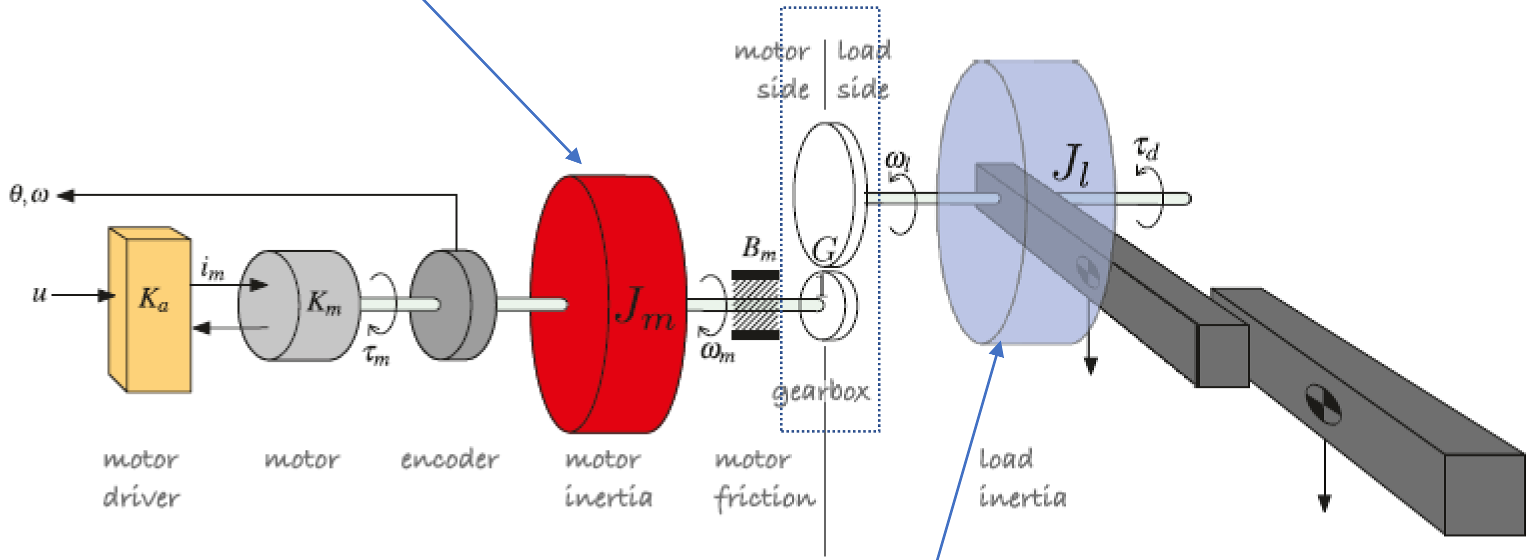
$$C_2 = -a_1m_2(a_2 + c_2)\sin q_2$$

$$g = (a_1m_1 + a_1m_2 + c_1m_1)\cos q_1 + (a_2m_2 + c_2m_2)\cos(q_1 + q_2)$$

In summary, the effect of joint motion in a series of mechanical links is nontrivial. The motion of any joint is affected by the motion of *all* the other joints and for a robot with many joints this becomes quite complex.

Gearbox

- There are two components of inertia *seen* by the motor. The first is due to the rotating part of the motor itself, its rotor. It is denoted *Jm* and is a constant intrinsic characteristic of the motor and the value is provided in the motor manufacturer’ s data sheet.



- The second component is the variable load inertia *Jl* which is the inertia of the driven link and all the other links that are attached to it. For joint *j* this is element *Mjj* of the configuration dependent inertia matrix

Modelling the Robot Joint

The complete motor drive comprises the motor to generate torque, the gearbox to amplify the torque and reduce the effects of the load, and an encoder to provide feedback of position and velocity. A schematic of such a device is shown in Fig. 9.6.

Collecting the various equations above we can write the torque balance on the motor shaft as

$$K_m K_a u - B'\omega - \tau_C'(\omega) - \frac{\tau_d(q)}{G} = J'\dot{\omega} \qquad (9.4)$$

where B', τ_C' and J' are the effective total viscous friction, Coulomb friction and inertia due to the motor, gearbox, bearings and the load

$$B' = B_m + \frac{B_l}{G^2}, \quad \tau_C' = \tau_{C,m} + \frac{\tau_{C,l}'}{G}, \quad J' = J_m + \frac{J_l}{G^2} \qquad (9.5)$$

In order to analyze the dynamics of Eq. 9.4 we must first linearize it, and this can be done simply by setting all additive constants to zero

$$J'\dot{\omega} + B'\omega = K_m K_a u$$

and then applying the Laplace transformation

$$sJ'\Omega(s) + B'\Omega(s) = K_m K_a U(s)$$

where $\Omega(s)$ and $U(s)$ are the Laplace transform of the time domain signals $\omega(t)$ and $u(t)$ respectively. This can be rearranged as a linear transfer function

Table 9.1. Relationship between load and motor referenced quantities for reduction gear ratio G

${}^{l}J = G^2\,{}^{m}J$
${}^{l}B = G^2\,{}^{m}B$
${}^{l}\tau_C = G\,{}^{m}\tau_C$
${}^{l}\tau = G\,{}^{m}\tau$
${}^{l}\omega = {}^{m}\omega/G$
${}^{l}\dot{\omega} = {}^{m}\dot{\omega}/G$

Modelling the Robot Joint (continued)

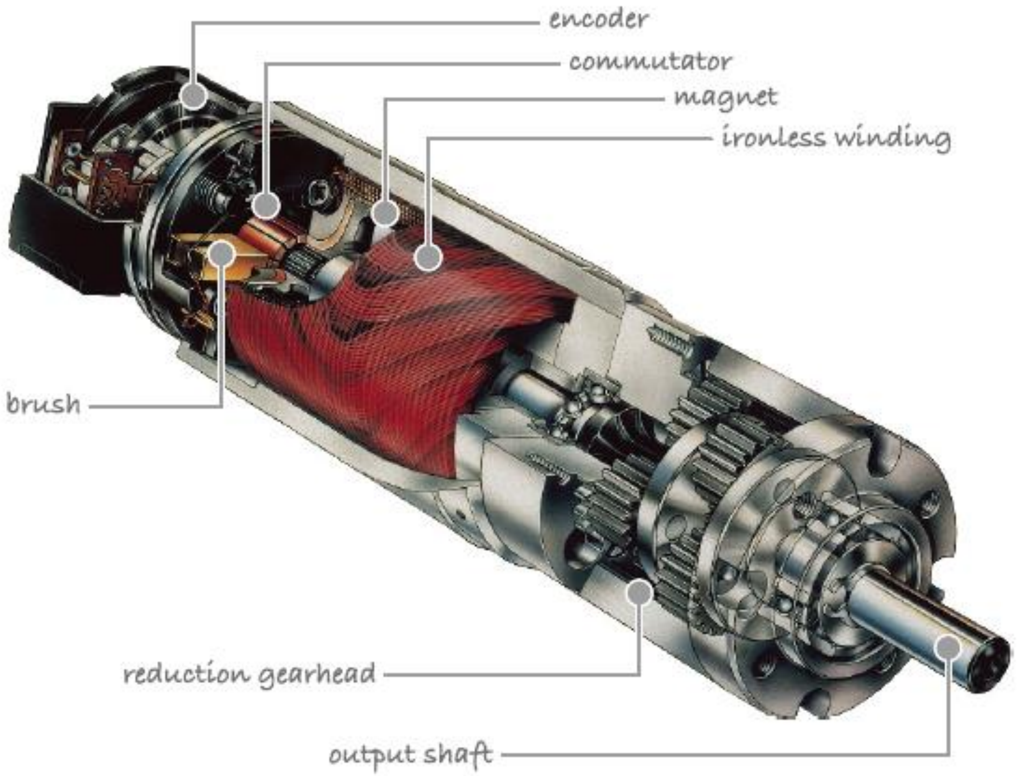


Fig. 9.6. Schematic of an integrated motor-encoder-gearbox assembly (courtesy of maxon precision motors, inc.)

Table 9.2. Motor and drive parameters for Puma 560 shoulder joint with respect to the motor side of the gearbox (Corke 1996b)

Parameter	Symbol	Value	Unit
Motor torque constant	K_m	0.228	N m A^{-1}
Motor inertia	J_m	200×10^{-6}	kg m^2
Drive viscous friction	B_m	817×10^{-6}	N m s rad^{-1}
Drive Coulomb friction	τ_C^+ τ_C^-	0.126 −0.709	N m N m
Gear ratio	G	107.815	
Maximum torque	τ_{max}	0.900	N m
Maximum speed	$\dot{q}_{max}$	165	rad s^{-1}

$$\frac{\Omega(s)}{U(s)} = \frac{K_m K_a}{J' s + B'}$$

relating motor speed to control input, and has a single pole at $s = -B' / J'$.

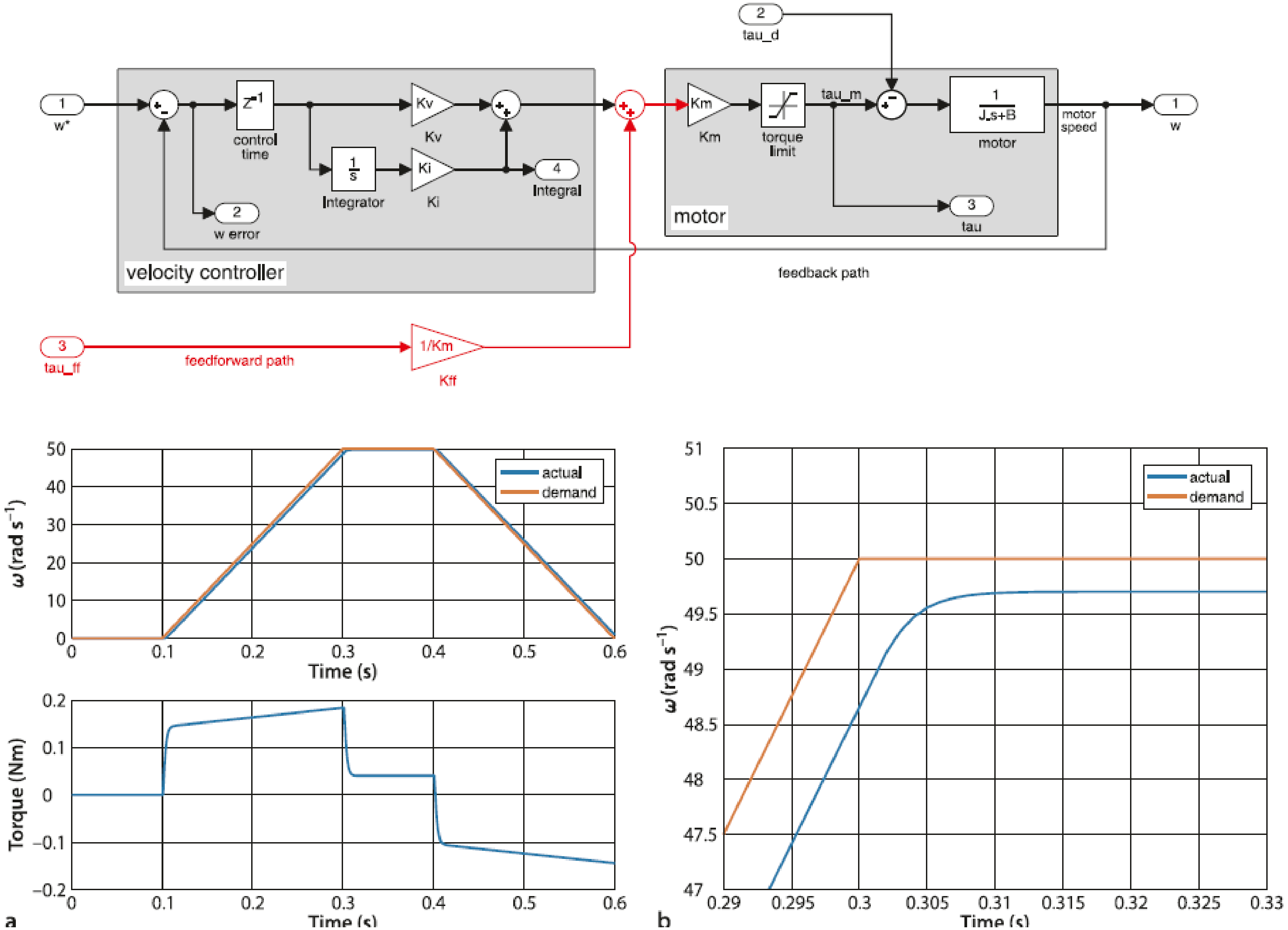
The mechanical pole.

Velocity Loop

A very common approach to controlling the position output of a motor is the nested control loop.

The outer loop is responsible for maintaining position and determines the velocity of the joint that will minimize position error.

The inner loop – the velocity loop – is responsible for maintaining the velocity of the joint as demanded by the outer loop.



Velocity Loop

There are three common approaches to counter this error. These two approaches are collectively referred to as disturbance rejection and are concerned with reducing the effect of an unknown disturbance.

The first, and simplest, is to increase the gain. This will reduce the tracking error but push the system toward instability and increase the overshoot.

The second approach, commonly used in industrial motor drives, is to add integral action – adding an integrator changes the system to Type 1 which has zero error for a constant input or constant disturbance.

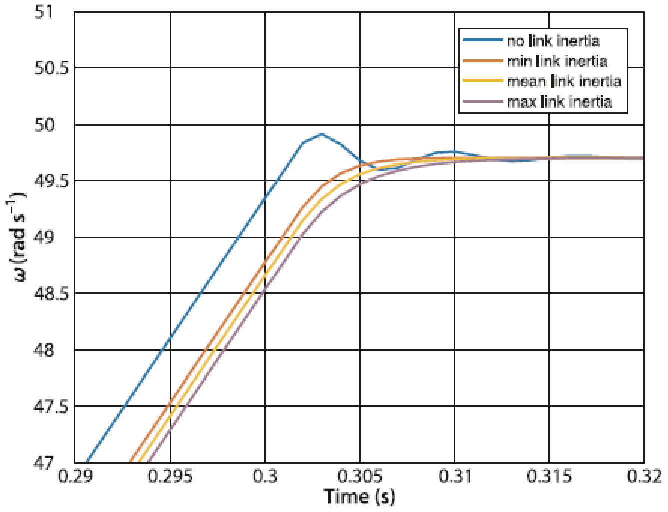


Fig. 9.10. Velocity loop response with a trapezoidal demand for varying inertia M_{22}

$$u^* = K_v\left(\dot{q}^* - \dot{q}\right)$$

$$u^* = \left(K_v + \frac{K_i}{s}\right)\left(\dot{q}^* - \dot{q}\right), \quad K_i > 0$$

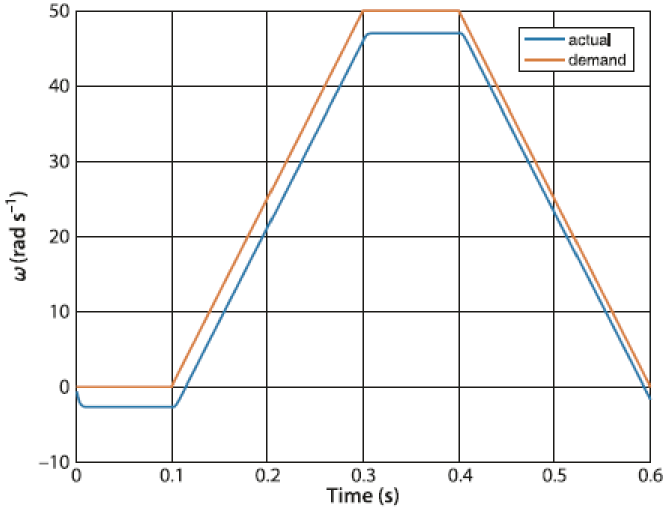


Fig. 9.11. Velocity loop response to a trapezoidal demand with a gravity disturbance of 20 N m

Velocity Loop

As always in engineering there are some tradeoffs.

The integral term can lead to increased overshoot so increasing Ki usually requires some compensating reduction of Kv. If the joint actuator is pushed to its performance limit, for instance the torque limit is reached, then the tracking error will grow with time since the motor acceleration will be lower than required.

The integral of this increasing error will grow leading to a condition known as integral windup.

The third approach is therefore to predict the disturbance and cancel it out – a strategy known as torque feedforward control

$$u^* = K_v\left(\dot{q}^* - \dot{q}\right)$$

$$u^* = \left(K_v + \frac{K_i}{s}\right)\left(\dot{q}^* - \dot{q}\right), \quad K_i > 0$$

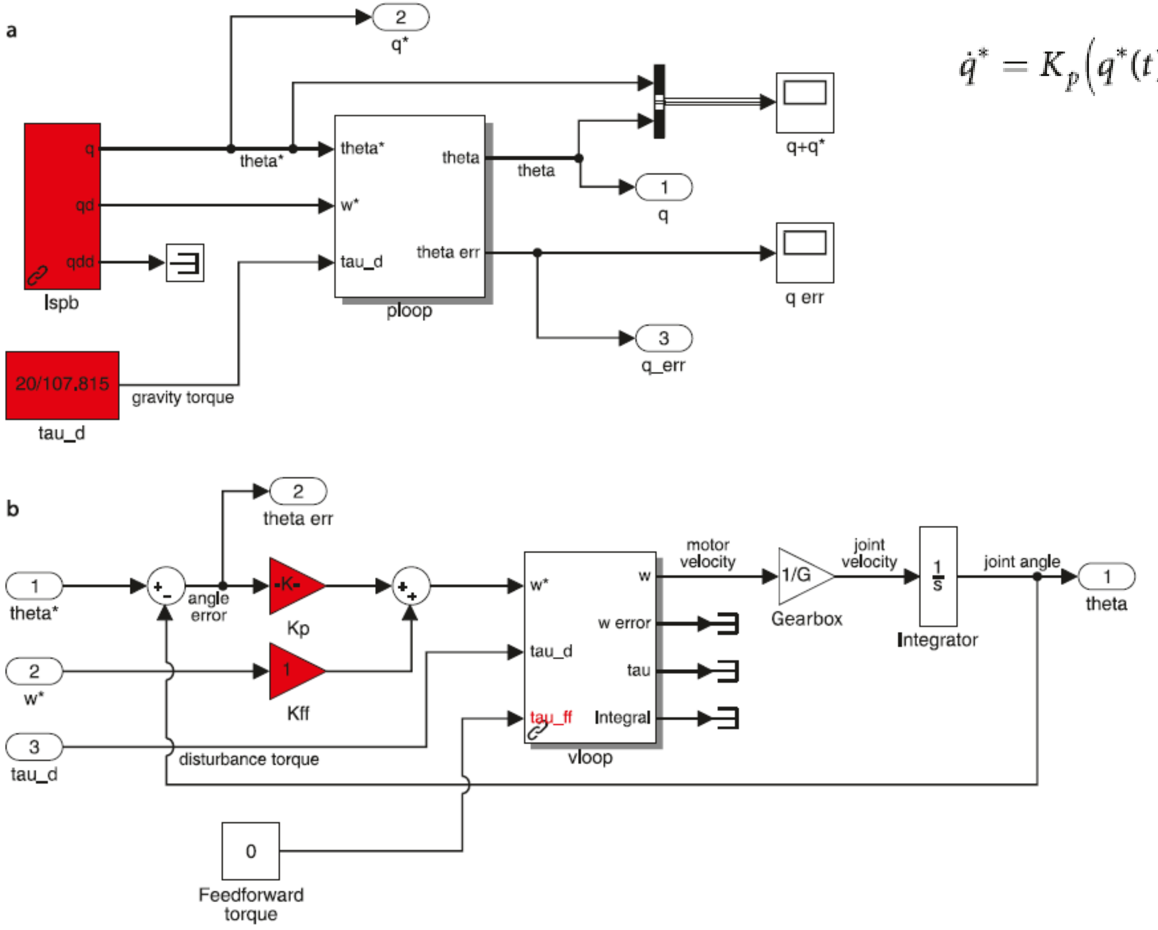
Position Control Loop

The outer loop is responsible for maintaining position.

The proportional Controller compute the desired speed of the motor based on the error between actual and demanded position.

The steady state error can be reduced at least two ways:

- Adding an integrator to the position loop – making it a proportional-integral controller but this
- Use velocity feedforward control – we add the desired velocity to the output of the proportional control loop



$$\dot{q}^* = K_p\left(q^*(t) - q\right)$$

Rigid-Body Equations of Motion

The series of links dynamic behavior can be written elegantly and concisely as a set of coupled differential equations in matrix form:

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

where q, $\dot{q}$ and $\ddot{q}$ are respectively the vector of generalized joint coordinates, velocities and accelerations, M is the joint-space inertia matrix, C is the Coriolis and centripetal coupling matrix, F is the friction force, G is the gravity loading, and Q is the vector of generalized actuator forces associated with the generalized coordinates q. The last term gives the joint forces due to a wrench W applied at the end-effector and J is the manipulator Jacobian. This equation describes the manipulator rigid-body dynamics and is known as the inverse dynamics – given the pose, velocity and acceleration it computes the required joint forces or torques.

Obtained using:

- Newton's second law and Euler's equation of motion,
- A Lagrangian energy-based approach.

The recursive Newton-Euler algorithm

- A very efficient
- Starts at the base and working outward
- Adds the velocity and acceleration of each joint in order to determine the velocity and acceleration of each link.

Rigid-Body Equations of Motion

The torque on joint 1 is that needed to overcome friction which always opposes the motion.

Torques need to be exerted on joints 2, 3 and 4 to oppose the gyroscopic effects (centripetal and Coriolis forces)

Such torques are referred to as velocity coupling torques since the rotational velocity of one joint has induced a torque on several other joints.

9.2.5 Effect of Payload

Any real robot has a specified maximum payload which is dictated by two dynamic effects. The first is that a mass at the end of the robot will increase the inertia *experienced* by the joint motors and which reduces acceleration and dynamic performance. The second is that mass generates a weight force which all the joints need to support. In the worst case the increased gravity torque component might exceed the rating of one or more motors. However even if the rating is not exceeded there is less torque available for acceleration which again reduces dynamic performance.

9.2.1 Gravity Term

the gravity term is generally the dominant term in the equation and is present even when the robot is stationary or moving slowly.

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

9.2.2 Inertia Matrix

Inertia is the natural tendency of objects in motion to stay in motion and objects at rest to stay at rest, unless a force causes the velocity to change

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

The joint-space inertia is a positive definite, and is therefore symmetric. It is a function of the manipulator configuration.

9.2.3 Coriolis Matrix

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

The off-diagonal terms Ci,j represent coupling of joint j velocity to the generalized force acting on joint i.

9.2.4 Friction

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

For most electric drive robots friction is the next most dominant joint force after gravity.

9.2.6 Base Force

A moving robot exerts a wrench on its base – its weight as well as reaction forces and torques as the arm moves around. This wrench is returned as an optional output argument of the `rne` method, for example

9.2.7 Dynamic Manipulability

In Sect. 8.2.2 we discussed a kinematic measure of manipulability, that is, how well configured the robot is to achieve velocity in any Cartesian direction. The force ellipsoid of Sect. 8.5.2 describes how well the manipulator is able to accelerate in different Cartesian directions but is based on the kinematic, not dynamic, parameters of the robot arm.

9.3 Forward Dynamics

To determine the motion of the manipulator in response to the forces and torques applied to its joints we require the forward dynamics or integral dynamics. Rearranging the equations of motion Eq. 9.8 we obtain the joint acceleration

$$\ddot{q} = M^{-1}(q)\left(Q - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q) - J(q)^T W\right) \tag{9.10}$$

and M is always invertible. This function is computed by the `accel` method of the `SerialLink` class

Rigid-Body Dynamics Compensation

Inertia and coupling torques can be computed according to Eq. 9.8 given knowledge of joint angles, joint velocities and accelerations, and the inertial parameters of the links.

These torques can be integrated into the control law using one of two model-based approaches: feedforward control, and computed torque control.

The feedforward term linearizes the nonlinear dynamics about the operating point

9.4.1 Feedforward Control

The torque feedforward controller shown in Fig. 9.20 is given by

$$
\begin{aligned}
Q^* &= \underbrace{M(q^*)\ddot{q}^* + C(q^*,\dot{q}^*)\dot{q}^* + F(\dot{q}^*) + G(q^*)}_{\text{feedforward}} + \underbrace{\left\{K_v(\dot{q}^* - \dot{q}^\#) + K_p(q^* - q^\#)\right\}}_{\text{feedback}} \\
&= \mathcal{D}^{-1}(q^*,\dot{q}^*,\ddot{q}^*) + \left\{K_v(\dot{q}^* - \dot{q}^\#) + K_p(q^* - q^\#)\right\} \qquad (9.11)
\end{aligned}
$$

where K_p and K_v are the position and velocity gain (or damping) matrices respectively, and $\mathcal{D}^{-1}(\cdot)$ is the inverse dynamics function. The gain matrices are typically diagonal. The feedforward term provides the joint forces required for the desired manipulator state $(q^*, \dot{q}^*, \ddot{q}^*)$ and the feedback term compensates for any errors due to uncertainty

9.4.2 Computed Torque Control

The computed torque controller is shown in Fig. 9.21. It belongs to a class of controllers known as inverse dynamic control. The principle is that the nonlinear system is cascaded with its inverse so that the overall system has a constant unity gain. In practice the inverse is not perfect so a feedback loop is required to deal with errors.

The computed torque control is given by

$$
\begin{aligned}
Q &= M(q)\left\{\ddot{q}^* + K_v(\dot{q}^* - \dot{q}^\#) + K_p(q^* - q^\#)\right\} + C(q^*,\dot{q}^*)\dot{q}^* + F(\dot{q}^*) + G(q^*) \\
&= \mathcal{D}^{-1}\left(q^*,\dot{q}^*,\left(\ddot{q}^* + K_v(\dot{q}^* - \dot{q}^\#) + K_p(q^* - q^\#)\right)\right) \qquad (9.13)
\end{aligned}
$$

where K_p and K_v are the position and velocity gain (or damping) matrices respectively, and $\mathcal{D}^{-1}(\cdot)$ is the inverse dynamics function.

Rigid-Body Dynamics Compensation

We can also consider that the feedforward term linearizes the nonlinear dynamics about the operating point.

$$M(q^*)\ddot{e} + K_v\dot{e} + K_p e = 0 \qquad (9.12)$$

Unlike feedforward control, the joint errors are uncoupled and their dynamics are therefore independent of manipulator configuration. In the case of model error there will be some coupling between axes, and the right-hand side of Eq. 9.14 will be a nonzero forcing function.

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \qquad (9.14)$$

error $e = q^* - q^\#$

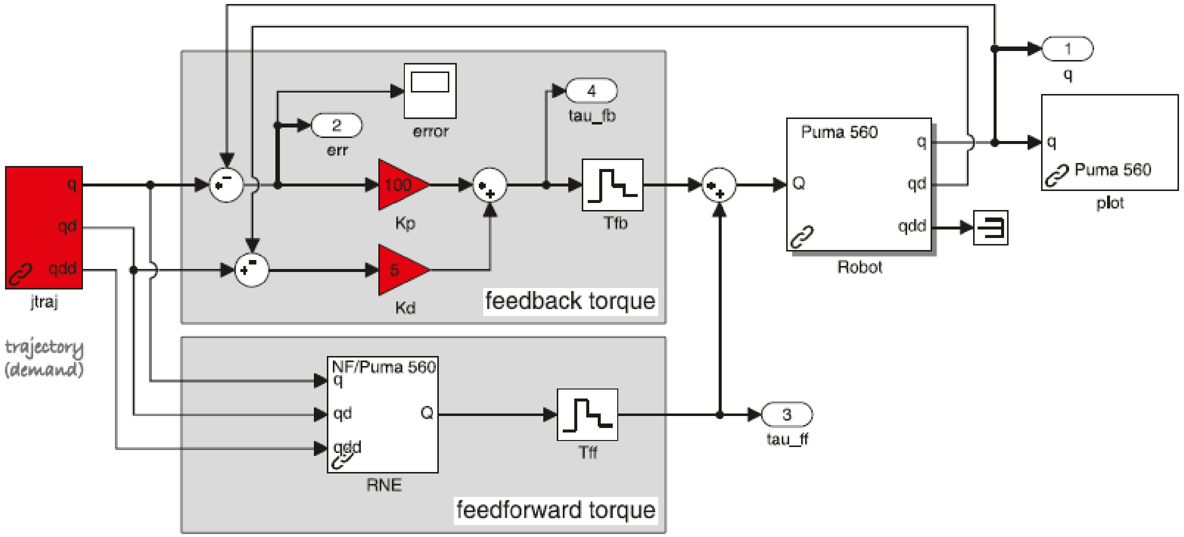
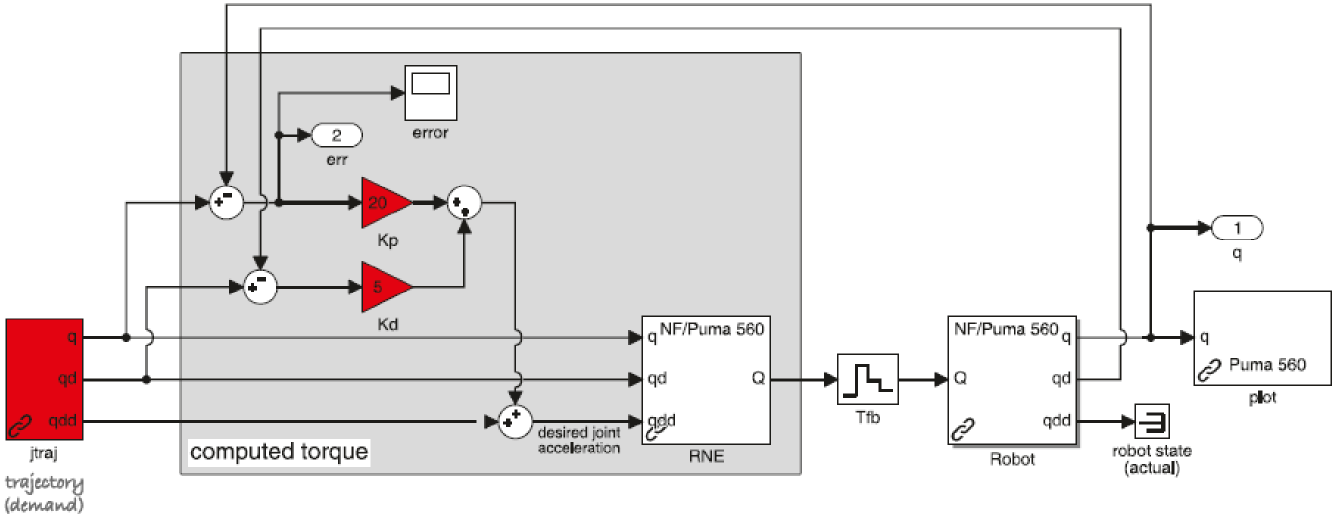


Fig. 9.20. The Simulink model `sl_fforward` for Puma 560 with torque feedforward control. The blocks with the staircase icons are zero-order holds

Fig. 9.21. Robotics Toolbox example `sl_ctorque`, computed torque control



9.4.3 Operational Space Control

The control strategies so far have been posed in terms of the robot's joint coordinates – its configuration space. Equation 9.8 describes the relationship between joint position, velocity, acceleration and applied forces or torques. However we can also express the dynamics of the end-effector in the Cartesian operational space where we consider the end-effector as a rigid body with inertia that actuator and disturbance forces and torques act on. We can reformulate Eq. 9.8 in operational space as

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{W} \tag{9.15}$$

$$Q = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T W$$

$${}^0\nu = {}^0J(q)\dot{q}$$

where $\boldsymbol{x} \in \mathbb{R}^6$ is the manipulator Cartesian pose and $\boldsymbol{\Lambda}$ is the end-effector inertia which is subject to a gyroscopic and Coriolis force μ and gravity load p and an applied control wrench W. These operational space terms are related to those we have already discussed by

$$\begin{aligned}
\boldsymbol{\Lambda}(\boldsymbol{x}) &= J(q)^{-T} J(q) J(q)^{-1} \\
\mu(\boldsymbol{x}, \dot{x}) &= J(q)^{-T} C(q,\dot{q}) - \Lambda(q)\dot{J}(q)\dot{q} \\
p(\boldsymbol{x}) &= J(q)^{-T} g(q)
\end{aligned}$$

the acceleration in operational space. This can be found by taking the time derivative of our original Jacobian equation.

Additional info: https://studywolf.wordpress.com/2013/09/17/robot-control-4-operation-space-control/

Main Concepts to Remember

- Robots can be controlled with Independent Joint Control or Rigid-Body Dynamics Compensation
- The Rigid-Body Equations of Motion provides insights on how the robot behaves
- Forward Dynamics allow use to model the robotic arm

9.6 Wrapping Up

In this Chapter we discussed approaches to robot manipulator control. We started with the simplest case of independent joint control, and explored the effect of disturbance torques and variation in inertia, and showed how feedforward of disturbances such as gravity could provide significant improvement in performance. We then learned how to model the forces and torques acting on the individual links of a serial-link manipulator. The equations of motion or inverse dynamics compute the joint forces required to achieve particular joint velocity and acceleration. The equations have terms corresponding to inertia, gravity, velocity coupling, friction and externally applied forces. We looked at the significance of these terms and how they vary with manipulator configuration and payload. The equations of motion provide insight into important issues such as how the velocity or acceleration of one joint exerts a disturbance force on other joints which is important for control design. We then discussed the forward dynamics which describe how the configuration evolves with time in response to forces and torques applied at the joints by the actuators and by external forces such as gravity. We extended the feedforward notion to full model-based control using torque feedforward, computed torque and operational-space controllers. Finally we discussed series-elastic actuators where a compliant element between the robot motor and the link enables force control and people-safe operation.