UiO : Department of Technology Systems
University of Oslo

8. Motion control – Centralized control

Kim Mathiassen

Lecture overview

- Centralized control joint control
  - PD control with gravity compensation
  - Inverse dynamics control
- Centralized operational space control
  - PD control with gravity compensation
  - Inverse dynamics control

Motivation

- When large operational speeds are required or when direct drive is emplyed nonlinear couplings strongly influence the system performance
- A manipulator is not a set of n decoupled systems, but a multivariate system with n inputs and n outputs
- Finding nonlinear multivariable control laws u to track trajectories

$$\boldsymbol{\tau} = \boldsymbol{u} = \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{G}_i \boldsymbol{v}_c \tag{8.17}$$

8.5.1 PD control with gravity compensation

- Using Lyapunov direct method to determine the control input
- System state is $[\,\widetilde{\boldsymbol{q}}^T \quad \dot{\boldsymbol{q}}^T\,]^T$
  – where $\widetilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$
- Lyapunov function candidate

$$V(\dot{\boldsymbol{q}}, \widetilde{\boldsymbol{q}}) = \underbrace{\frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}}_{\text{Kinetic energy}} + \underbrace{\frac{1}{2}\widetilde{\boldsymbol{q}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{q}}}_{\text{Potential energy}} > 0 \qquad \forall \dot{\boldsymbol{q}}, \widetilde{\boldsymbol{q}} \neq \boldsymbol{0} \qquad (8.47)$$

Deriving control law

- Differentiating the Lyapunov candidate function yields

$$\dot{V} = \frac{1}{2}\dot{\boldsymbol{q}}^T\left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T\boldsymbol{F}\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\left(\boldsymbol{u} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_P\widetilde{\boldsymbol{q}}\right). \qquad (8.49)$$

Negative definite

Null (7.49)

- Need to select u so last term becomes negative semi-definit

Deriving control law

- Intuitive approach

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P \widetilde{\boldsymbol{q}}, \qquad (8.50)$$

- Yields negative semi-definite $\dot{V}$ since

$$\dot{V} = 0 \qquad\qquad \dot{\boldsymbol{q}} = \boldsymbol{0}, \forall \widetilde{\boldsymbol{q}}.$$

Deriving control law

- Using

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P\widetilde{\boldsymbol{q}} - \boldsymbol{K}_D\dot{\boldsymbol{q}}, \tag{8.51}$$

- Yields

$$\dot{V} = -\dot{\boldsymbol{q}}^T(\boldsymbol{F} + \boldsymbol{K}_D)\dot{\boldsymbol{q}}, \tag{8.52}$$

- This corresponds to nonlinear compensation action of gravitational terms with a linear PD controller
- K_D increases the system response time

Remarks

- The derivative term is crucial for direct-drive manipulators, as mechanical viscous damping (F) is practically null
- The control law requires online computation of g(q)
- The system reaches an equilibrium posture

$$\dot{V} \equiv 0 \text{ only if } \dot{\boldsymbol{q}} \equiv \boldsymbol{0}.$$

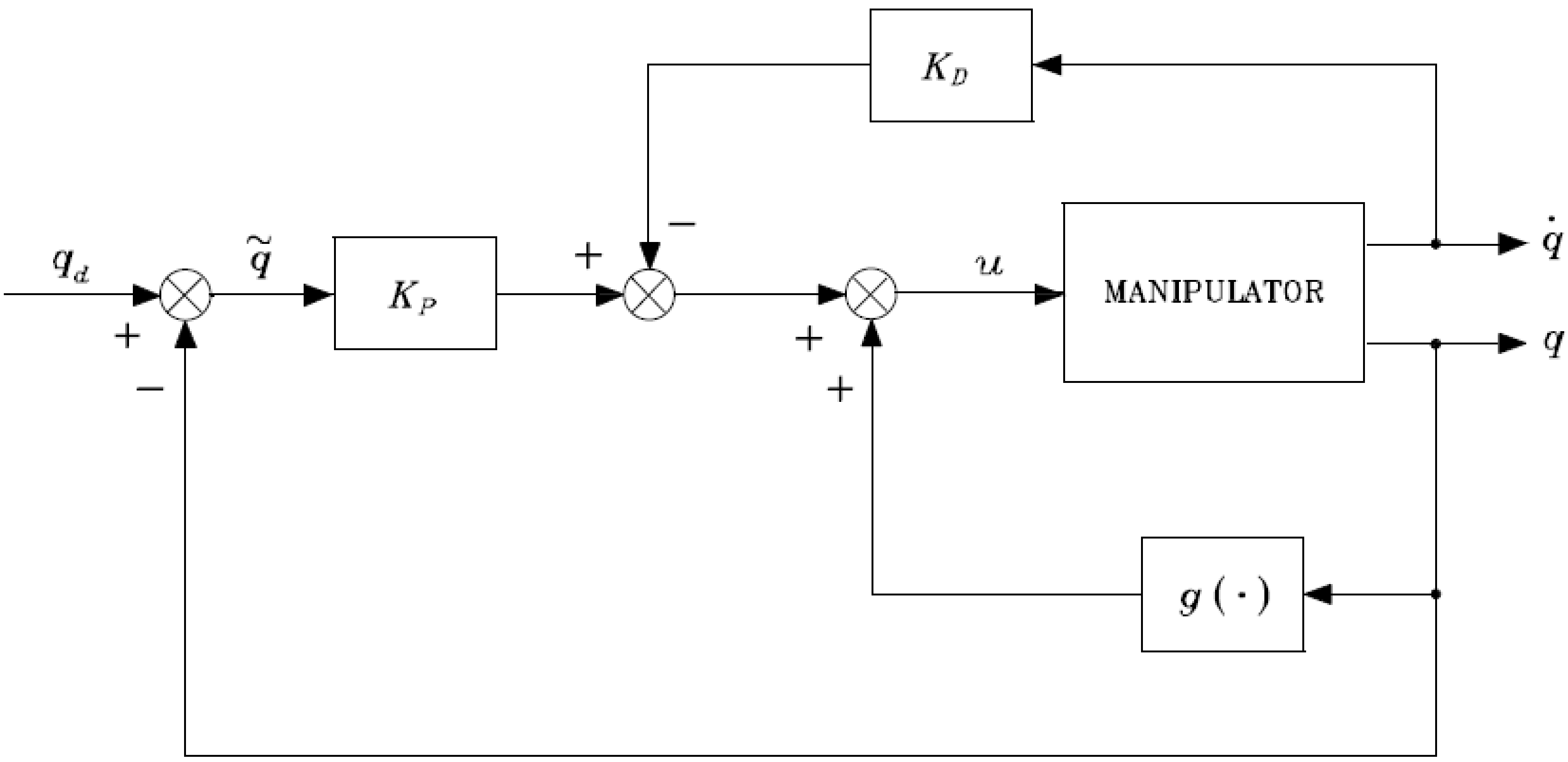
- Given the system dynamics

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P\widetilde{\boldsymbol{q}} - \boldsymbol{K}_D\dot{\boldsymbol{q}}. \qquad (8.53)$$

- Inserting $\dot{\boldsymbol{q}} \equiv \boldsymbol{0},\ \ddot{\boldsymbol{q}} \equiv \boldsymbol{0}$ to find the equilibrium posture yields

$$\boldsymbol{K}_P\widetilde{\boldsymbol{q}} = \boldsymbol{0} \qquad\qquad \widetilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q} \equiv \boldsymbol{0}$$

Block diagram of PD controller with gravity compensation



8.5.2 Inverse dynamics control

- The PD controller with gravity compensation has only joint position as input
- Now we consider tracking a joint space trajectory
- Rewriting the dynamic equation as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{u}, \tag{8.55}$$

- where

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}). \tag{8.56}$$

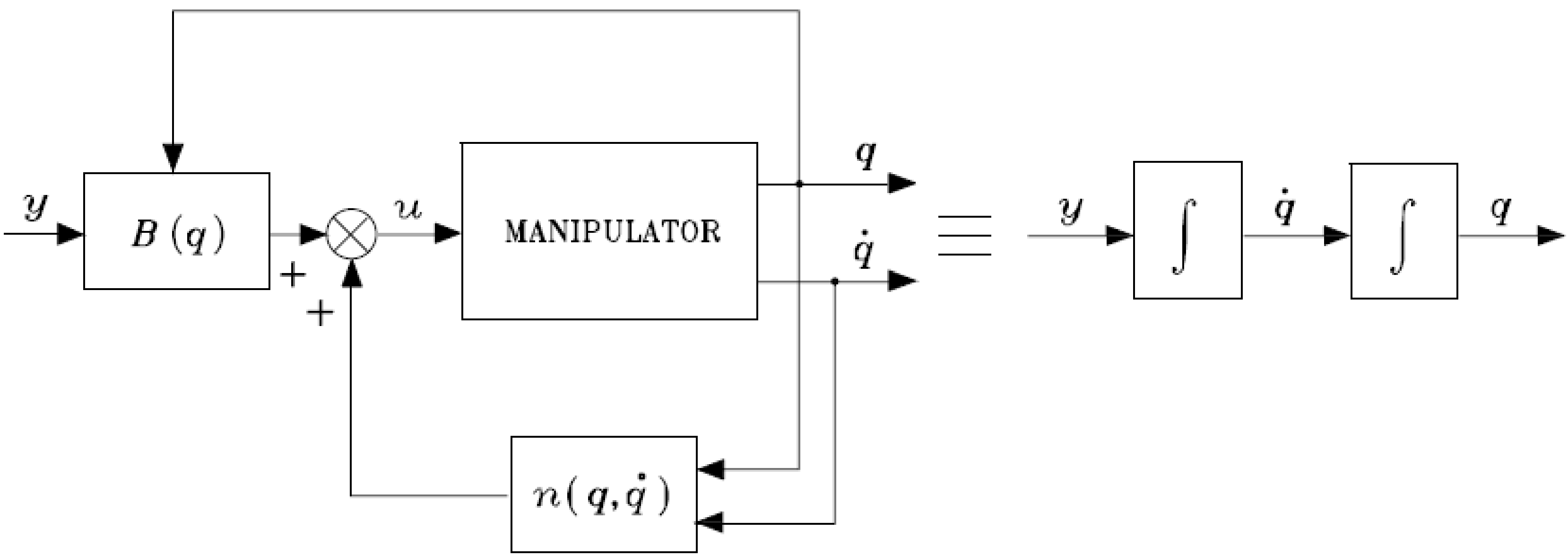
Exact linearization of the system dynamics

- The idea is to find u which makes the system a linear system
- B(q) has full rank and can be inverted for any manipulator configuration
- Selecting u as

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{8.57}$$

- Yields $\ddot{\boldsymbol{q}} = \boldsymbol{y}$
- y represents a new input vector

Exact linearization



Remarks

- The nonlinear control law (8.57) is termed inverse dynamics control, as it it based on the inverse dynamics of the manipulator
- The new system is linear and decoupled with respect to y

Control law for decoupled system

- Choosing the control law

$$y = -K_P q - K_D \dot{q} + r \qquad (8.58)$$

- Yields a system of second order equations

$$\ddot{q} + K_D \dot{q} + K_P q = r \qquad (8.59)$$

- Choosing K_D and K_P as

$$K_P = \mathrm{diag}\{\omega_{n1}^2, \ldots, \omega_{nn}^2\} \qquad K_D = \mathrm{diag}\{2\zeta_1\omega_{n1}, \ldots, 2\zeta_n\omega_{nn}\},$$

- This gives a decoupled asymptotically stable system

Tracking the trajectory

- Tracking of the trajectory q_td(t) is ensured by choosing

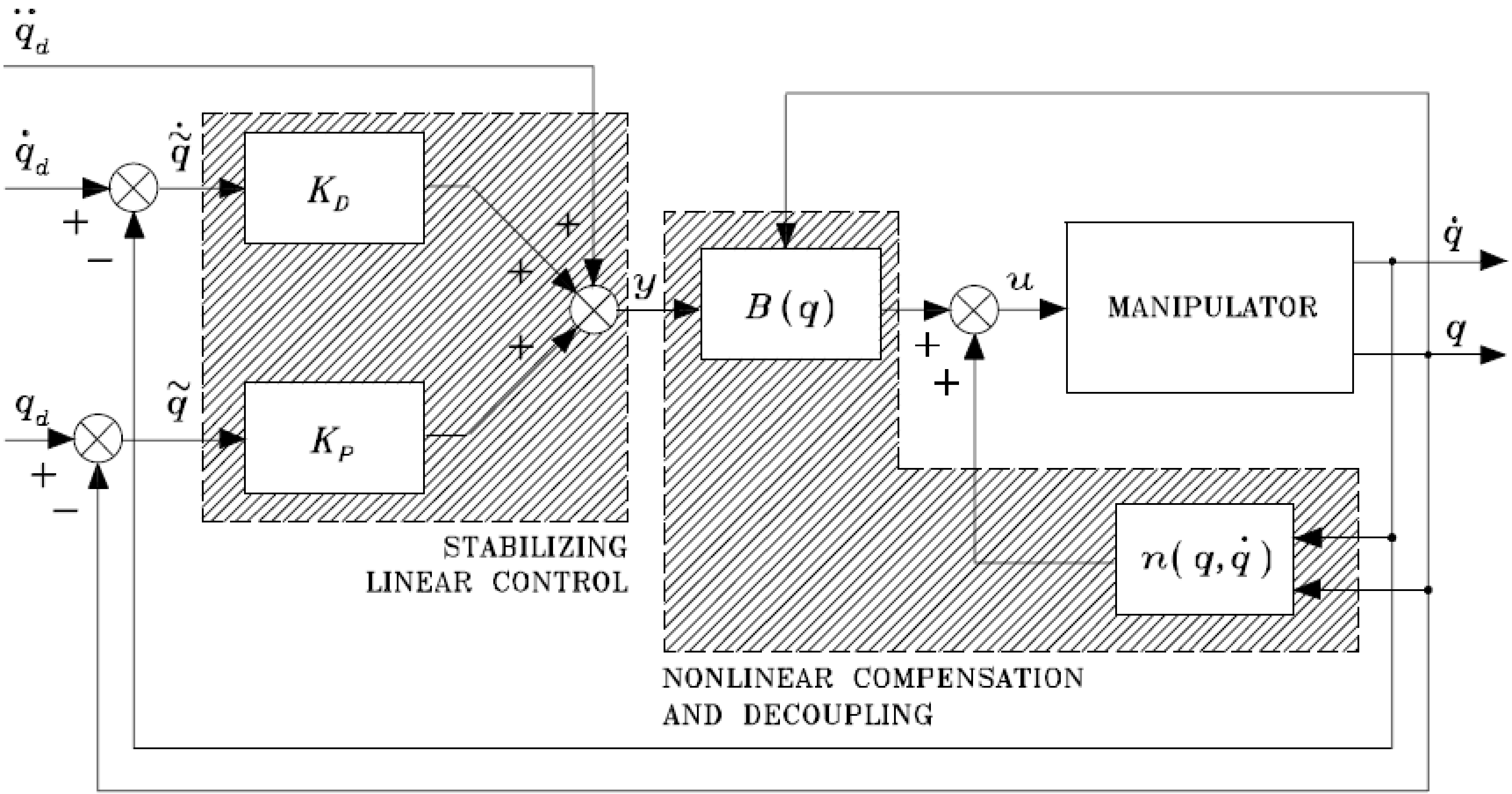
$$\boldsymbol{r} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D \dot{\boldsymbol{q}}_d + \boldsymbol{K}_P \boldsymbol{q}_d. \tag{8.60}$$

- This yields the error dynamics

$$\ddot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_D \dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_P \widetilde{\boldsymbol{q}} = \boldsymbol{0} \tag{8.61}$$

- Error only occures because of initial conditions

Block diagram



Remarks

- Outer control loop design is simplified as it operates on a linear time-invariant system
- Must compute B(q) and n(q,qd) online
- Controller based on the assumtion of perfect cancelation
- All model must be perfectly known
- Sampling time must be in the order of milliseconds to ensure that operation assumption of continuous time domain
- May calculate the dominating terms to address real-time issues

8.5.3 Robust control

- Address the issues with the previous controller, i.e. the model may be imperfect, expressed by

$$\boldsymbol{u} = \widehat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{8.62}$$

- The error in the estimates (i.e. uncertainty) is represented by

$$\widetilde{\boldsymbol{B}} = \widehat{\boldsymbol{B}} - \boldsymbol{B} \qquad \widetilde{\boldsymbol{n}} = \widehat{\boldsymbol{n}} - \boldsymbol{n} \tag{8.63}$$

- This yields the dynamic system

$$\boldsymbol{B}\ddot{\boldsymbol{q}} + \boldsymbol{n} = \widehat{\boldsymbol{B}}\boldsymbol{y} + \widehat{\boldsymbol{n}} \tag{8.64}$$

Error dynamics

- Rearraanging the equiation yields

$$\ddot{q} = y + (B^{-1}\widehat{B} - I)y + B^{-1}\widetilde{n} = y - \eta \tag{8.65}$$

- where

$$\eta = (I - B^{-1}\widehat{B})y - B^{-1}\widetilde{n}. \tag{8.66}$$

- Using similar control as in inverse dynamics control

$$y = \ddot{q}_d + K_D(\dot{q}_d - \dot{q}) + K_P(q_d - q),$$

- Leads to

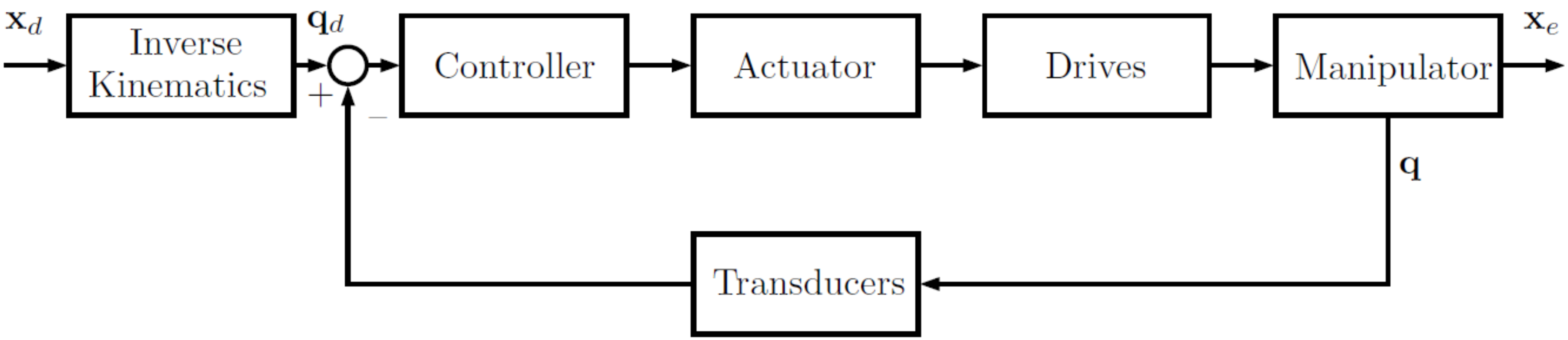
$$\ddot{\widetilde{q}} + K_D\dot{\widetilde{q}} + K_P\widetilde{q} = \eta. \tag{8.67}$$

Remarks

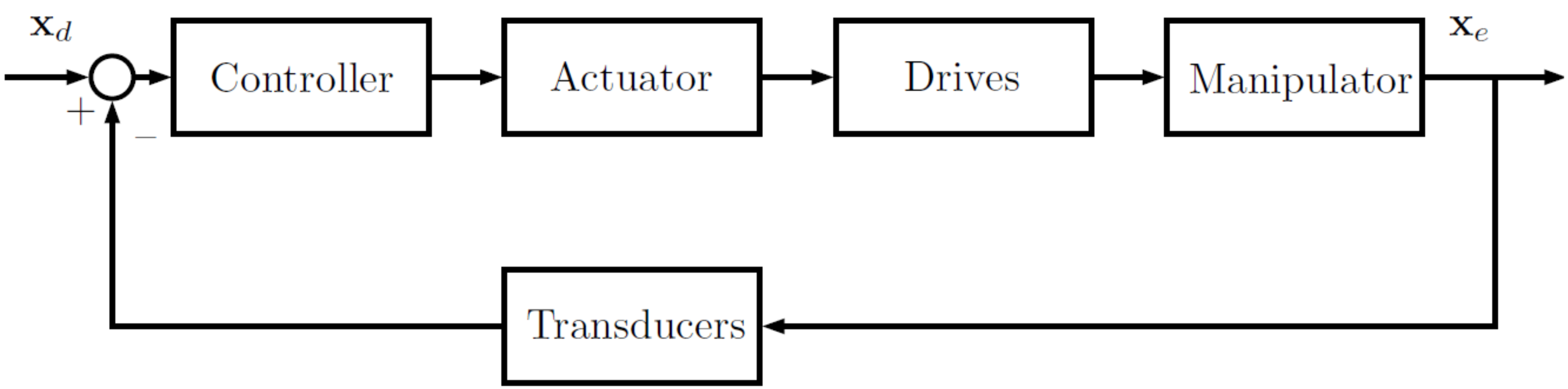
$$\ddot{\tilde{q}} + K_D \dot{\tilde{q}} + K_P \tilde{q} = \eta. \tag{8.67}$$

- The system is coupled and nonlinear, since η is a nonlinear funtion of the errors (position and velocity)
- A linear PD controller is no longer sufficient for tracking a trajectory given uncertainties
- It is possible to add a robustness term to compensate for some of the errors

8.6 Operational space control



Joint space control



Operational space control

Motivation

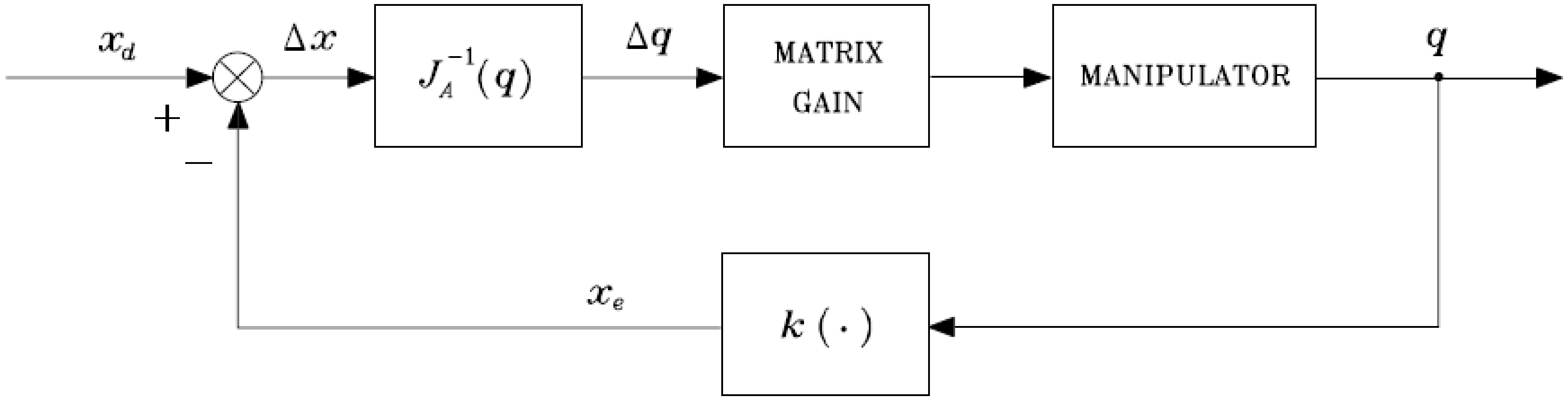
- Joint space approaches requires calculation of inverse kinematics and inverse first and second order differential kinematics
- Industrial robots often calculate the joint positions through inverse kinematics, and then perform numerical differentiation to compute velocities and acceleration
- A different approach is to have a control scheme that is developed directly in the operational space
- Joint space control suffice only for motion in free space, and the following scheme will constitute the premise for the force/position control of the next chapter

$$\dot{x}_e = J_A(q)\dot{q}$$

Jacobian inverse control

$$K\,\Delta q = \tau$$

- Intuitively behaves like a mechanical system with a generalized n-dimentional spring in joint space, whose stiffness is determined by the feedback matrix gain
- Δx can also be viewed as a velocity which controls the system to minimize the error

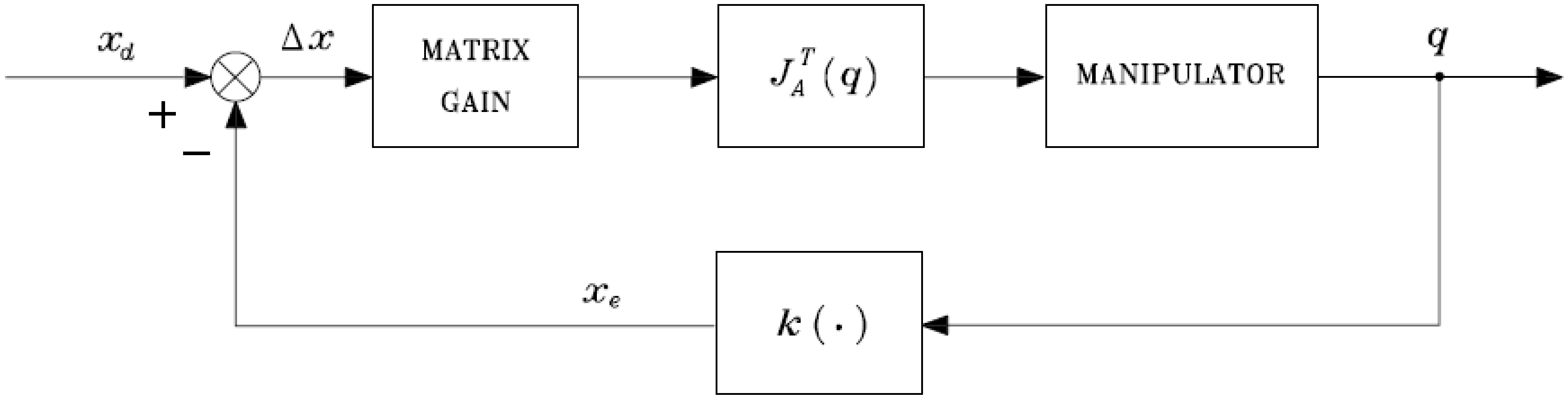


$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma}_e$$

Jacobian transpose control

$$K\ \Delta x = \gamma_e$$

- The output of the matrix gain block can be viewed as a force generated by a spring in the operational space
- The operational space force is then transformes to jointspace forces



8.6.2 PD control with gravity compensation

- Given a constant end effector pose, we would like a control law that tends the error asymptocically to zero

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}_e \tag{8.106}$$

- Choosing the following Lyapunov function candidate

$$V(\dot{\boldsymbol{q}}, \widetilde{\boldsymbol{x}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\widetilde{\boldsymbol{x}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{x}} > 0 \qquad \forall \dot{\boldsymbol{q}}, \widetilde{\boldsymbol{x}} \neq \boldsymbol{0}, \tag{8.107}$$

- Time differentiating

$$\dot{V} = \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\widetilde{\boldsymbol{x}}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{x}}.$$

Deriving control law

- Since $\dot{\boldsymbol{x}}_d = \mathbf{0}$,

$$\dot{\widetilde{\boldsymbol{x}}} = -\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Then

$$\dot{V} = \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\widetilde{\boldsymbol{x}}. \qquad (8.108)$$

- By recalling

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \qquad (8.7)$$

$$\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \qquad (7.47)$$

$$\dot{\boldsymbol{q}}^T \boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = 0; \qquad (7.49)$$

- yields

$$\dot{V} = -\dot{\boldsymbol{q}}^T \boldsymbol{F}\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \left(\boldsymbol{u} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\widetilde{\boldsymbol{x}}\right). \qquad (8.109)$$

Deriving control law

- Choosing the control law

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\widetilde{\boldsymbol{x}} - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{8.110}$$

- Yields

$$\dot{V} = -\dot{\boldsymbol{q}}^T\boldsymbol{F}\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{8.111}$$

- This gives the equilibrium posture

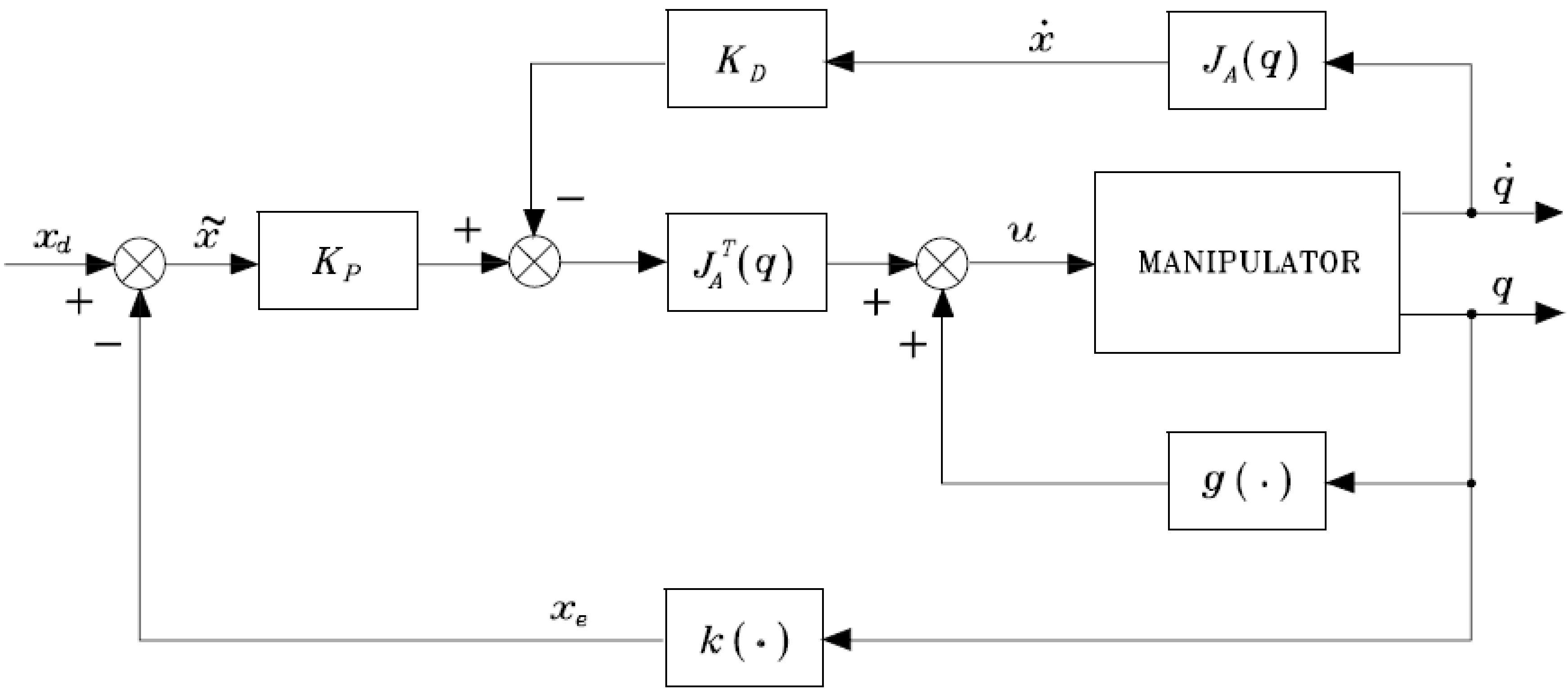
$$\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\widetilde{\boldsymbol{x}} = \boldsymbol{0}. \tag{8.112}$$

- Under the assumption of full rank Jacobian

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}_e = \boldsymbol{0},$$

Block diagram

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\widetilde{\boldsymbol{x}} - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{8.110}$$



8.6.3 Inverse dynamics control

- Now we consider following a operational space trajectory
- Recall that

$$B(q)\ddot{q} + n(q, \dot{q}) = u,$$

$$u = B(q)y + n(q, \dot{q})$$

- Transformed the system to a system of two double integrators

$$\ddot{q} = y. \tag{8.113}$$

- Now we will design a new control input y which will track the trajectory $x_d(t)$

Deriving control law

- We start by using the second order differential equation

$$\ddot{\boldsymbol{x}}_e = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

- Choosing the control law

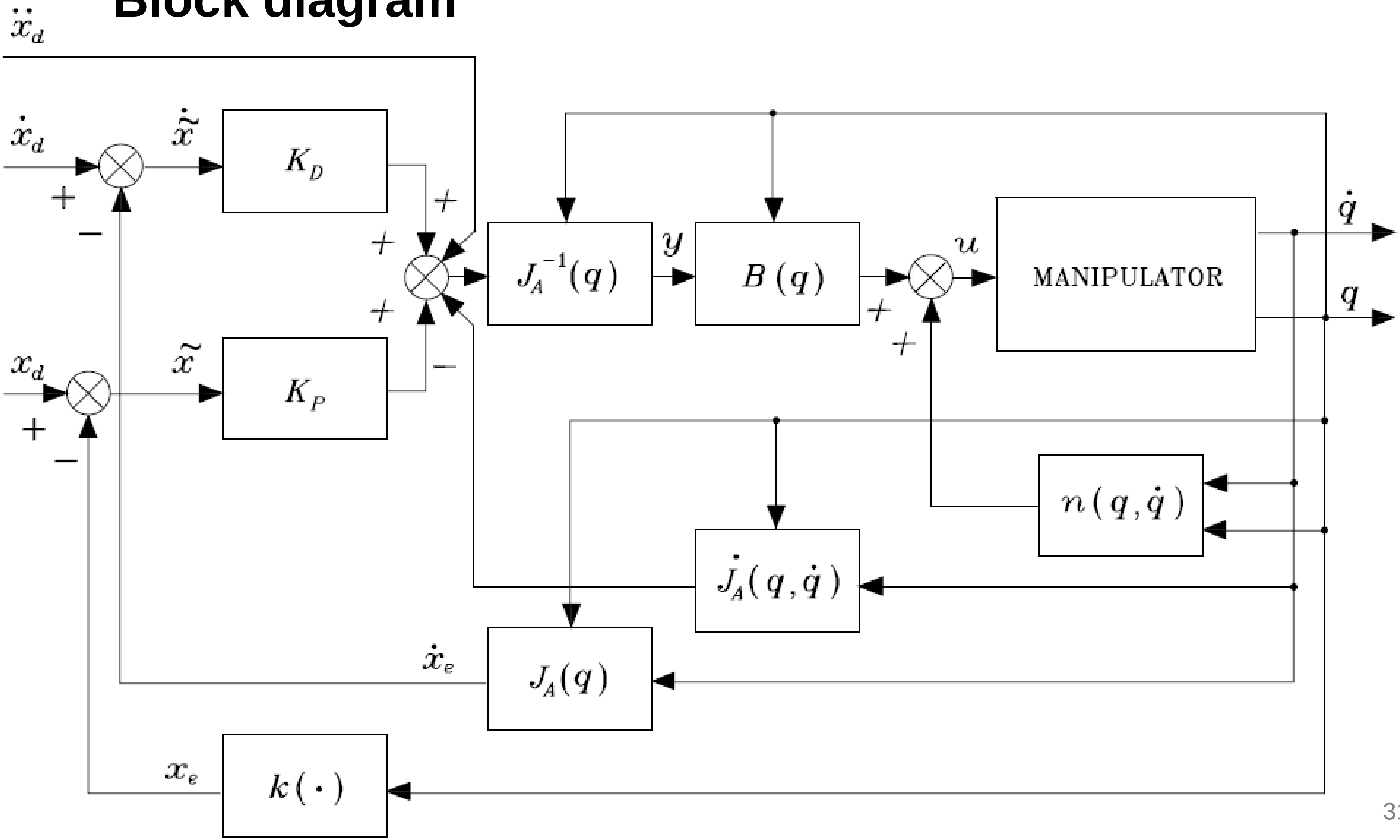
$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) \qquad (8.114)$$

- This yields the error dynamics in operational space

$$\ddot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{x}} = \boldsymbol{0} \qquad (8.115)$$

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\left(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right) \qquad (8.114)$$

Block diagram



Remarks for operational space control

- Operational space control always require computation of Jacobian
- This means that controlling in operational space is more complex that joint space because of singularities and redundancy
- The Jacobian transpose scheme might get stuck because of singularities
- For the Jacobian inverse scheme a singular Jacobian leads to infinite control input
- Redundancy handling must be incorporated into the control loop

Summary

	Joint space	Operational space
Setpoint controller	PD control with gravity compensation	PD control with gravity compensation (Jacobian transpose control)
Trajectory tracking	Inverse dynamics control	Inverse dynamics control (Jacobian inverse control)

Exercises

1. Prove (8.111) based on (8.107)
2. Prove (8.115)