ELEC4410

Control Systems Design

Lecture 20: Scaling and MIMO State Feedback Design

Julio H. Braslavsky

julio@ee.newcastle.edu.au

School of Electrical Engineering and Computer Science
The University of Newcastle

Outline

- A Note about Scaling
- MIMO State Feedback Design
  - Cyclic Design
  - MIMO Regulation and Tracking
- MIMO Observers

A Note About Scaling

- State space is the preferred model for LTI systems, especially with higher order models. However, even with state-space models, accurate results are not guaranteed, because of the **finite-word-length arithmetic** of the computer.

A Note About Scaling

- State space is the preferred model for LTI systems, especially with higher order models. However, even with state-space models, accurate results are not guaranteed, because of the **finite-word-length arithmetic** of the computer.
- When calculations are performed in a computer, each arithmetic operation is affected by *roundoff error*, since machine hardware can only represent a subset of the real numbers.

A Note about Scaling

Normalisation:

A well-conditioned problem is usually a prerequisite for obtaining accurate results. One should generally normalize or scale the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ of a system to improve their numerical conditioning.

A Note about Scaling

Normalisation:

> A well-conditioned problem is usually a prerequisite for obtaining accurate results. One should generally normalize or scale the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ of a system to improve their numerical conditioning.

- Normalization also allows meaningful statements to be made about the degree of controllability and observability of the various inputs and outputs.

A Note about Scaling

A set of matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ can be normalized using diagonal scaling matrices $\mathbf{N}_u$, $\mathbf{N}_x$ and $\mathbf{N}_y$ to scale $\mathbf{u}, \mathbf{x}$, and $\mathbf{y}$,

$$\mathbf{u} = \mathbf{N}_u\,\mathbf{u}_n, \quad \mathbf{x} = \mathbf{N}_x\,\mathbf{x}, \quad \mathbf{y} = \mathbf{N}_y\,\mathbf{y}_n$$

so that the normalised system is

$$\begin{aligned} \dot{\mathbf{x}}_n &= \mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n \\ \mathbf{y}_n &= \mathbf{C}_n\mathbf{x}_n + \mathbf{D}_n\mathbf{u}_n \end{aligned} \quad \text{where} \quad \begin{aligned} \mathbf{A}_n &= \mathbf{N}_x^{-1}\mathbf{A}\mathbf{N}_x, \quad \mathbf{B}_n = \mathbf{N}_x^{-1}\mathbf{B}\mathbf{N}_u \\ \mathbf{C}_n &= \mathbf{N}_y^{-1}\mathbf{C}\mathbf{N}_x, \quad \mathbf{D}_n = \mathbf{N}_y^{-1}\mathbf{D}\mathbf{N}_u \end{aligned}$$

One criterion for the normalisation is to use the maximum expected range of each of the input, state, and output variables, e.g., say ± 10 Volts.

If possible, choose scaling based upon physical insight to the problem at hand.

A Note about Scaling

MATLAB provides the function `ssbal` to obtain automatic scaling of the state space vector. Specifically,

```
G=ss(A,B,C,D);
Gn=ssbal(G);
```

uses `balance` to compute a diagonal similarity transformation $\mathbf{N}_x$ such that

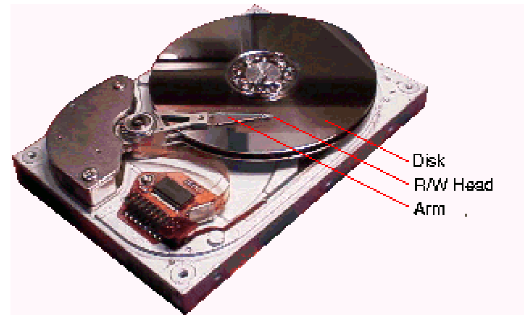
$$\begin{bmatrix} \mathbf{N}_x^{-1}\mathbf{A}\mathbf{N}_x & \mathbf{N}_x^{-1}\mathbf{B} \\ \mathbf{C}\mathbf{N}_x & 0 \end{bmatrix}$$

has equal row and column norms.

Such diagonal scaling is an economical way to compress the numerical range and improve the conditioning of subsequent state-space computations.

A Note about Scaling

Example (Hard Disk Drive Problem).
Consider the model of HDD system:



$$G(s) = \frac{K_0 \omega_r^2}{(s^2 + 2\xi\omega_r s + \omega_r^2)s^2}$$

which is realised in the CCF as

$$\dot{x}(t) = \begin{bmatrix} -2\xi\omega_r & -\omega_r^2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 0 & K_0\omega_r^2 \end{bmatrix} x(t).$$

A Note about Scaling

Data from a real HDD give the parameters

$$K_0 = 1.507 \times 10^4, \quad \xi = 0.1, \quad \omega_r = 2\pi \times 3400.$$

With these parameters the matrices in CCF are numerically ill-conditioned, and MATLAB yields for the controllability matrix

```
>> rank(ctrb(A,B))
ans = 2
```

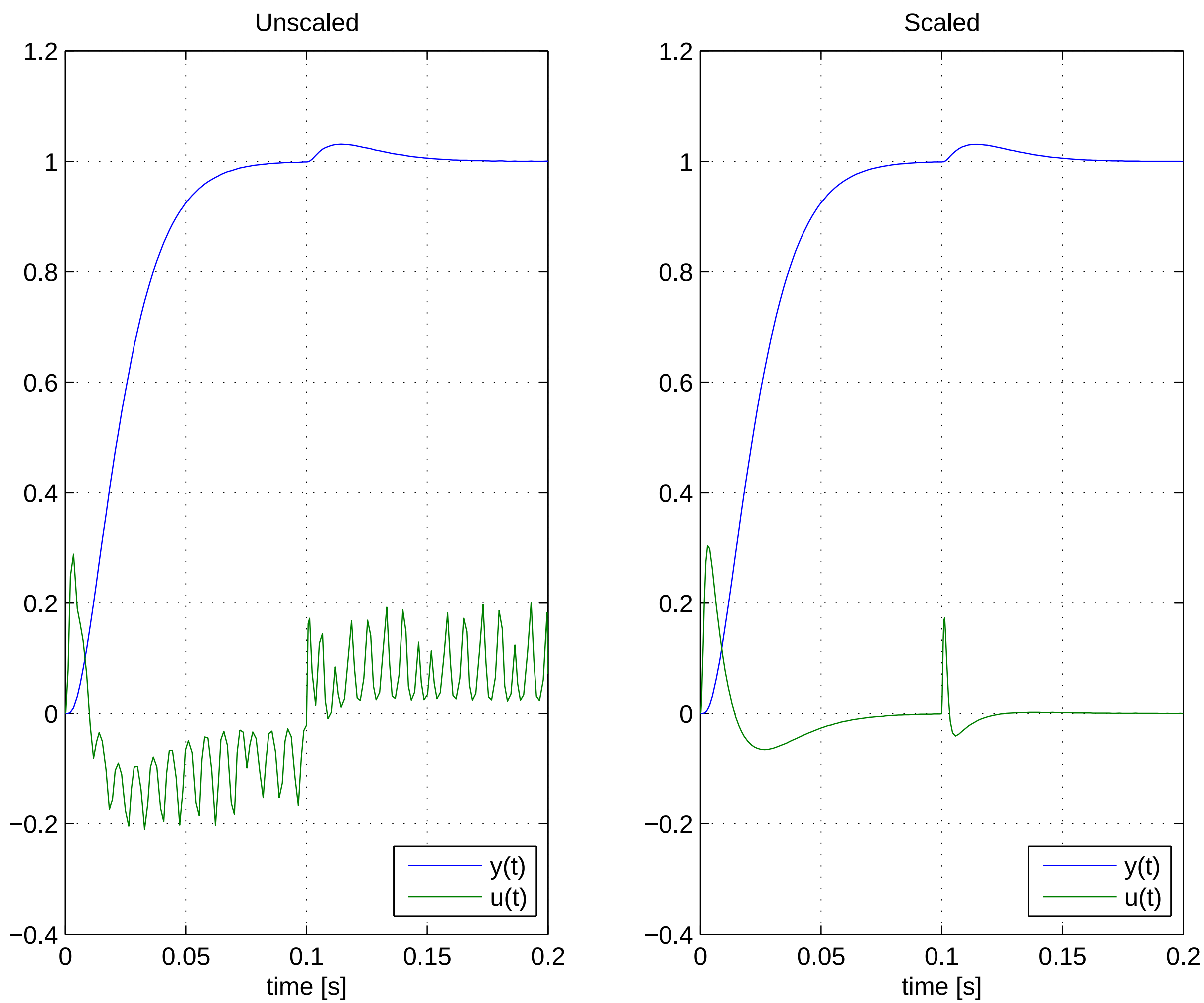
although the realisation **is** controllable by definition.

Scaling with MATLAB function `ssbal` gives the correct answer:

```
>> G=ss(A,B,C,D);
>> Gn=ssbal(G);
>> rank(ctrb(Gn.a,Gn.b))
ans = 4
```

A Note about Scaling

Also the closed-loop simulation results improve by scaling with `ssbal`



Outline

- A Note about Scaling
- MIMO State Feedback Design
  - Cyclic Design
  - MIMO Regulation and Tracking
- MIMO Observers

MIMO State Feedback Design

We now return to discuss MIMO LTI systems, this time, in the state space framework.

Recall from Lecture 10 that MIMO systems presented additional difficulties in the transfer function "language". The concepts of poles and zeros were more complicated, and control design, such as IMC design, turned out to be quite more messier than for SISO systems, particularly for nonsquare, possibly unstable plants.

The state space representation is particularly suited to MIMO systems. As we will see, there is no essential difference with the SISO procedures for state space control and observer design, even when the plant is nonsquare.

MIMO State Feedback Design

Before entering into the **design methods**, let's note that the results regarding controllability and eigenvalue assignability extend to the MIMO case.

MIMO State Feedback Design

Before entering into the **design methods**, let's note that the results regarding controllability and eigenvalue assignability extend to the MIMO case.

Theorem (Controllability and Feedback — MIMO). The pair $(\mathbf{A} - \mathbf{B}\mathbf{K}, \mathbf{B})$, for any $p \times n$ real matrix $\mathbf{K}$ is controllable if and only if the pair $(\mathbf{A}, \mathbf{B})$ is controllable.

MIMO State Feedback Design

Before entering into the **design methods**, let's note that the results regarding controllability and eigenvalue assignability extend to the MIMO case.

Theorem (Controllability and Feedback — MIMO). The pair $(\mathbf{A} - \mathbf{B}\mathbf{K}, \mathbf{B})$, for any $p \times n$ real matrix $\mathbf{K}$ is controllable if and only if the pair $(\mathbf{A}, \mathbf{B})$ is controllable.

Theorem (Eigenvalue assignment — MIMO). All eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ can be assigned arbitrarily (provided complex eigenvalues are assigned in conjugated pairs) by selecting a real constant $\mathbf{K}$ if and only if $(\mathbf{A}, \mathbf{B})$ is controllable.

MIMO State Feedback Design

A MIMO system in state space is described with the same formalism we have been using for SISO systems, i.e.,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) & x \in \mathbb{R}^n, u \in \mathbb{R}^p \\ y(t) &= Cx(t) & y \in \mathbb{R}^q \end{aligned}$$

When the system has p inputs, the state feedback gain K in a feedback law

$$u = -Kx = -\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ k_{p1} & k_{p2} & \cdots & k_{pn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

will have $p \times n$ parameters. That is, $K \in \mathbb{R}^{p \times n}$.

MIMO State Feedback Design

A MIMO system in state space is described with the same formalism we have been using for SISO systems, i.e.,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^p \\ y(t) &= Cx(t) \qquad\qquad\qquad\quad y \in \mathbb{R}^q \end{aligned}$$

When the system has p inputs, the state feedback gain K in a feedback law

$$u = -Kx = -p\left\{\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ k_{p1} & k_{p2} & \cdots & k_{pn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right\}n$$

will have $p \times n$ parameters. That is, $K \in \mathbb{R}^{p\times n}$.

MIMO State Feedback Design

A MIMO system in state space is described with the same formalism we have been using for SISO systems, i.e.,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \qquad & x \in \mathbb{R}^n, u \in \mathbb{R}^p \\ y(t) &= Cx(t) & y \in \mathbb{R}^q \end{aligned}$$

When the system has p inputs, the state feedback gain K in a feedback law

$$u = -Kx = -p\left\{ \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ k_{p1} & k_{p2} & \cdots & k_{pn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right\} n$$

will have $p \times n$ parameters. That is, $K \in \mathbb{R}^{p \times n}$.

Because the system evolution matrix A still has n eigenvalues, we have p times more degrees of freedom than necessary!

MIMO State Feedback Design

Example (Nonuniqueness of K in MIMO state feedback). As a simple MIMO system consider the second order system with two inputs

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(t)$$

The system has two eigenvalues at $s = 0$, and it is controllable, since $\mathbf{B} = \mathbf{I}$, so $\mathcal{C} = [\, \mathbf{B} \;\; \mathbf{A}\mathbf{B} \,]$ is full rank.

Let's consider the state feedback

$$u(t) = -\mathbf{K}x(t) = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} x(t)$$

Then the closed loop evolution matrix is

$$\mathbf{A} - \mathbf{B}\mathbf{K} = \begin{bmatrix} -k_{11} & -k_{12} \\ 1-k_{21} & -k_{22} \end{bmatrix}$$

MIMO State Feedback Design

Example (Continuation). Suppose that we would like to place both closed-loop eigenvalues at $s = -1$, i.e., the roots of the characteristic polynomial $s^2 + 2s + 1$. Then, one possibility would be to select

$$\begin{cases} k_{11} = 2 \\ k_{12} = 1 \\ k_{21} = 0 \\ k_{22} = 0 \end{cases} \Rightarrow A - BK = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix} \Rightarrow \text{eigenvalues at } s = -1$$

MIMO State Feedback Design

Example (Continuation). Suppose that we would like to place both closed-loop eigenvalues at $s = -1$, i.e., the roots of the characteristic polynomial $s^2 + 2s + 1$. Then, one possibility would be to select

$$\begin{cases} k_{11} = 2 \\ k_{12} = 1 \\ k_{21} = 0 \\ k_{22} = 0 \end{cases} \Rightarrow A - BK = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix} \Rightarrow \text{eigenvalues at } s = -1$$

But the alternative selection

$$\begin{cases} k_{11} = 1 \\ k_{12} = \text{free} \\ k_{21} = 1 \\ k_{22} = -1 \end{cases} \Rightarrow A - BK = \begin{bmatrix} -1 & k_{12} \\ 0 & -1 \end{bmatrix} \Rightarrow \text{also eigenvalues at } s = -1$$

MIMO State Feedback Design

Example (Continuation). Suppose that we would like to place both closed-loop eigenvalues at $s = -1$, i.e., the roots of the characteristic polynomial $s^2 + 2s + 1$. Then, one possibility would be to select

$$\begin{cases} k_{11} = 2 \\ k_{12} = 1 \\ k_{21} = 0 \\ k_{22} = 0 \end{cases} \Rightarrow A - BK = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix} \Rightarrow \text{eigenvalues at } s = -1$$

But the alternative selection

$$\begin{cases} k_{11} = 1 \\ k_{12} = \text{free} \\ k_{21} = 1 \\ k_{22} = -1 \end{cases} \Rightarrow A - BK = \begin{bmatrix} -1 & k_{12} \\ 0 & -1 \end{bmatrix} \Rightarrow \text{also eigenvalues at } s = -1$$

As we see, **there are infinitely many** possible selections of K that will give the same eigenvalues of $(A - BK)$! □

MIMO State Feedback Design

The "excess of freedom" in MIMO state feedback design could be a problem if we don't know how to best use it. . .

There are several ways to tackle the problem of selecting $\mathbf{K}$ from an infinite number of possibilities, among them

- Cyclic Design. Reduces the problem to one of a **single input**, so we can apply the known rules.

MIMO State Feedback Design

The "excess of freedom" in MIMO state feedback design could be a problem if we don't know how to best use it. . .

There are several ways to tackle the problem of selecting $\mathbf{K}$ from an infinite number of possibilities, among them

- Cyclic Design. Reduces the problem to one of a **single input**, so we can apply the known rules.
- Controller Canonical Form Design. Extends the Bass-Gura formula to MIMO.

MIMO State Feedback Design

The "excess of freedom" in MIMO state feedback design could be a problem if we don't know how to best use it...

There are several ways to tackle the problem of selecting $\mathbf{K}$ from an infinite number of possibilities, among them

- Cyclic Design. Reduces the problem to one of a **single input**, so we can apply the known rules.
- Controller Canonical Form Design. Extends the Bass-Gura formula to MIMO.
- Optimal Design. Computes the **best** $\mathbf{K}$ by optimising a suitable cost function.

MIMO State Feedback Design

The "excess of freedom" in MIMO state feedback design could be a problem if we don't know how to best use it. . .

There are several ways to tackle the problem of selecting $\mathbf{K}$ from an infinite number of possibilities, among them

- Cyclic Design. Reduces the problem to one of a **single input**, so we can apply the known rules.
- Controller Canonical Form Design. Extends the Bass-Gura formula to MIMO.
- Optimal Design. Computes the **best** $\mathbf{K}$ by optimising a suitable cost function.

We will discuss the Cyclic and Optimal designs.

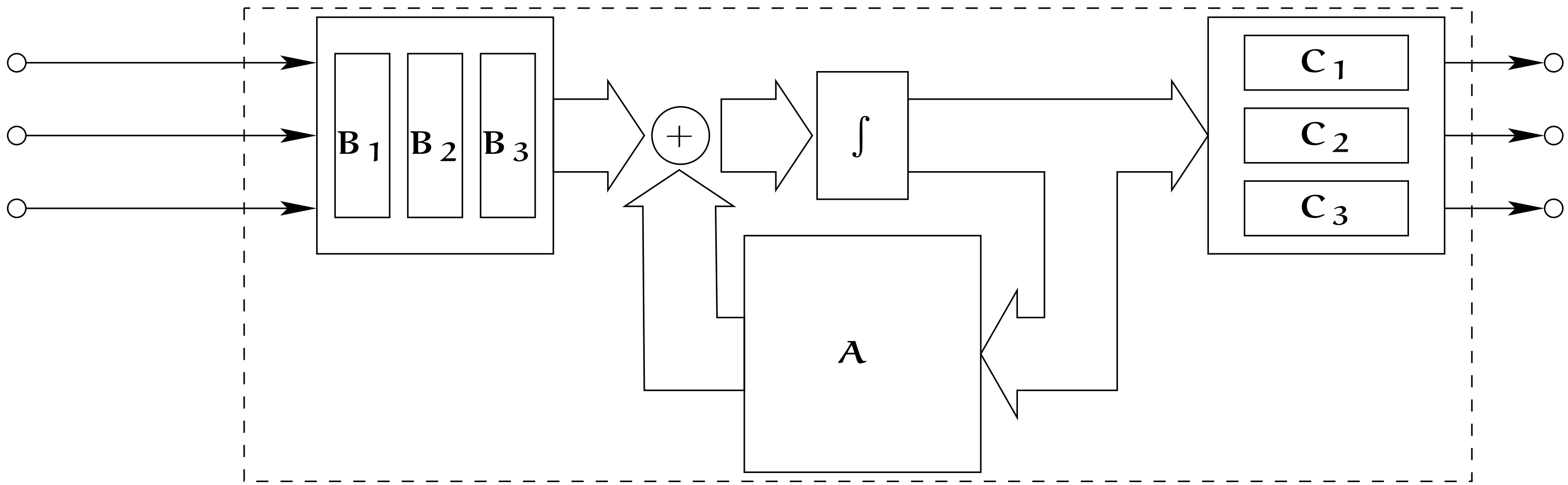
Outline

- A Note about Scaling
- MIMO State Feedback Design
  - Cyclic Design
  - MIMO Regulation and Tracking
- MIMO Observers

MIMO Cyclic State Feedback Design

MIMO Cyclic Design. In this design we change the multi-input problem into a single-input problem by creating a new input that is a linear combination of the inputs to the plant. This technique relies on the fact that

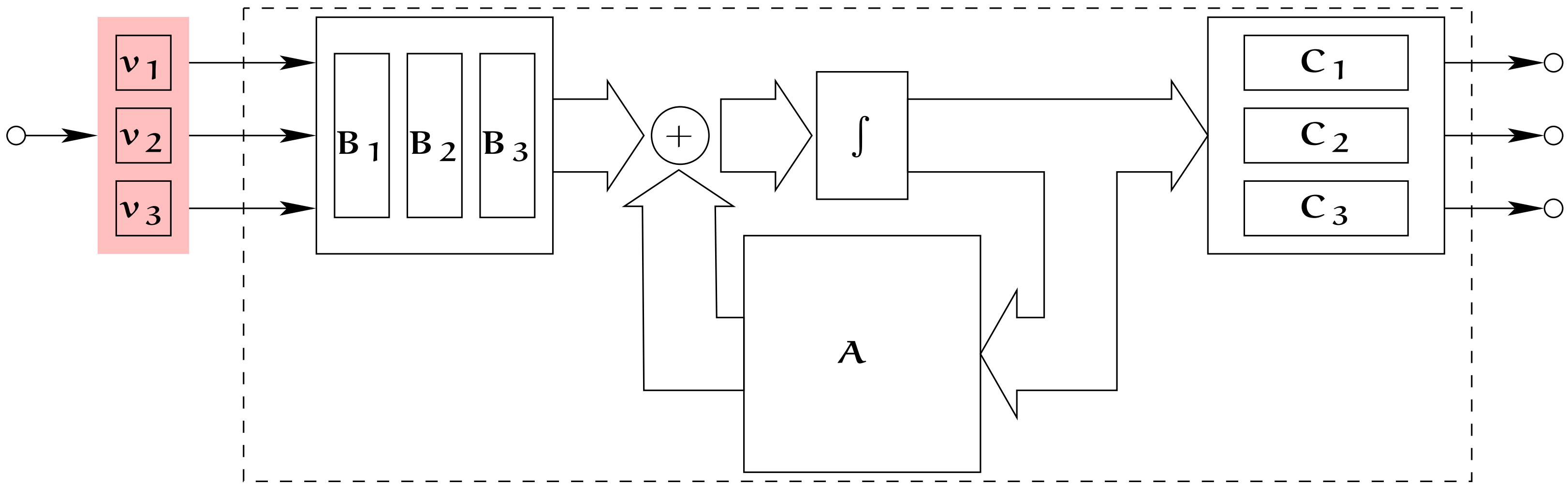
> If a state equation can be controlled by many inputs, then it can be controlled by one input.



MIMO Cyclic State Feedback Design

MIMO Cyclic Design. In this design we change the multi-input problem into a single-input problem by creating a new input that is a linear combination of the inputs to the plant. This technique relies on the fact that

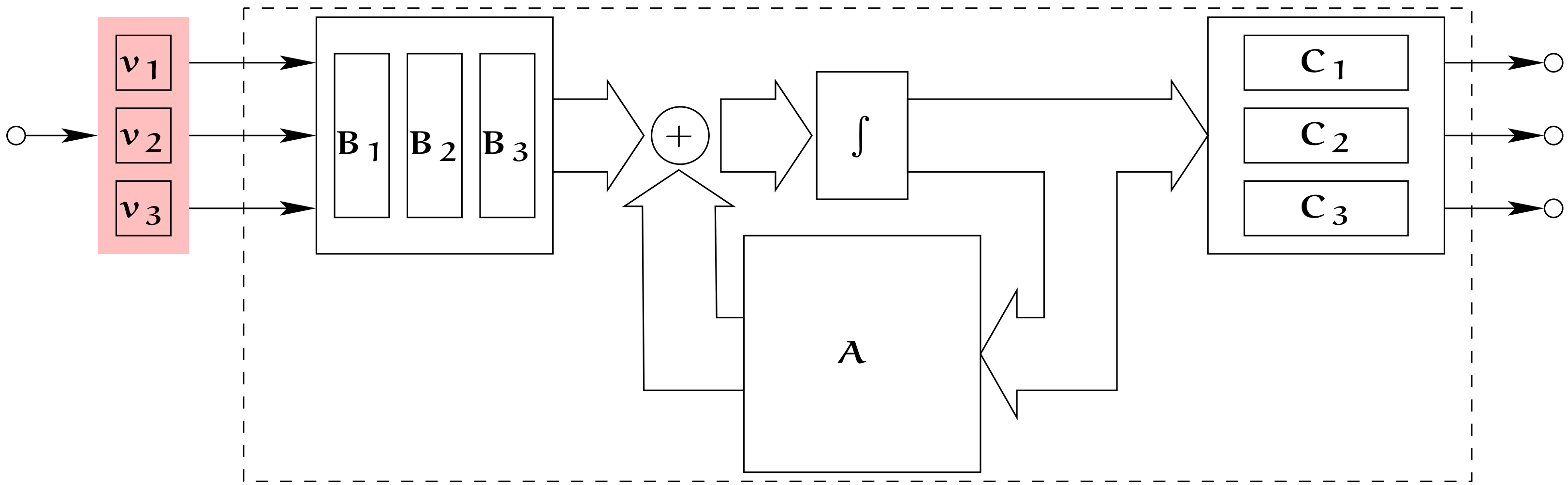
> If a state equation can be controlled by many inputs, then it can be controlled by one input.



MIMO Cyclic State Feedback Design

MIMO Cyclic Design. In this design we change the multi-input problem into a single-input problem by creating a new input that is a linear combination of the inputs to the plant. This technique relies on the fact that

> If a state equation can be controlled by many inputs, then it can be controlled by one input.



We need to define the **minimal polynomial** of a matrix, and what a **cyclic** matrix is, before proceeding with this technique.

MIMO Cyclic State Feedback Design

Recall that by the **Cayley Hamilton Theorem**, every matrix satisfies its **characteristic polynomial**, i.e.,

$$\text{if} \quad \Delta(s) = \mathbf{det}(sI - A) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_0,$$

$$\text{then} \quad \boxed{A^n + \alpha_1 A^{n-1} + \cdots + \alpha_0 I = 0.}$$

MIMO Cyclic State Feedback Design

Recall that by the **Cayley Hamilton Theorem**, every matrix satisfies its **characteristic polynomial**, i.e.,

$$\text{if} \quad \Delta(s) = \textbf{det}(sI - A) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_0,$$

$$\text{then} \quad \boxed{A^n + \alpha_1 A^{n-1} + \cdots + \alpha_0 I = 0.}$$

But the characteristic polynomial is not necessarily **the smallest degree monic polynomial a matrix may satisfy**. That polynomial is called the **minimal polynomial** of a matrix.

MIMO Cyclic State Feedback Design

Example (Minimal Polynomial of a Matrix). The characteristic polynomial of the matrix

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \text{is } \Delta(s) = (s+1)^2(s+2) = s^3 + 4s^2 + 5s + 2$$

Thus

$$A^3 + 4A^2 + 5A + 2I = 0.$$

But it could be verified that A also satisfies

$$\Delta_m(s) = (s+1)(s+2) = s^2 + 3s + 2,$$

which is the **smallest degree monic polynomial** A satisfies. That is the **minimal polynomial** of A. □

MIMO Cyclic State Feedback Design

A matrix A is **cyclic** if its characteristic polynomial **equals** its minimal polynomial.

MIMO Cyclic State Feedback Design

A matrix A is **cyclic** if its characteristic polynomial **equals** its minimal polynomial.

- **Fact:** The characteristic polynomial of a matrix equals its minimal polynomial if and only if in its Jordan form each eigenvalue is associated to one and only one Jordan block.

MIMO Cyclic State Feedback Design

A matrix A is **cyclic** if its characteristic polynomial **equals** its minimal polynomial.

- **Fact:** The characteristic polynomial of a matrix equals its minimal polynomial if and only if in its Jordan form each eigenvalue is associated to one and only one Jordan block.

Example.

$$A_1 = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 1 & 0 \\ 0 & 0 & \lambda_2 & 1 \\ 0 & 0 & 0 & \lambda_2 \end{bmatrix} \qquad A_2 = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 1 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & \lambda_2 \end{bmatrix}$$

- A_1 is cyclic: λ_1 and λ_2 associated to only one Jordan block each.
- A_2 is *not* cyclic: λ_1 associated to one Jordan block but λ_2 associated to two (one of order 1 and one of order 2).

MIMO Cyclic State Feedback Design

Notice that if a matrix has no repeated eigenvalues, then **it is cyclic**, since all the eigenvalues will be distinct and necessarily each associated to just one Jordan block (of order 1).

MIMO Cyclic State Feedback Design

Notice that if a matrix has no repeated eigenvalues, then **it is cyclic**, since all the eigenvalues will be distinct and necessarily each associated to just one Jordan block (of order 1).

Theorem (Controllability with p inputs and controllability with 1 input). If the n-dimensional p-input pair (A, B) is controllable and if A is cyclic, then for **almost any** $p \times 1$ vector V, the single-input pair (A, BV) is controllable.

MIMO Cyclic State Feedback Design

Notice that if a matrix has no repeated eigenvalues, then **it is cyclic**, since all the eigenvalues will be distinct and necessarily each associated to just one Jordan block (of order 1).

Theorem (Controllability with p inputs and controllability with 1 input). If the n-dimensional p-input pair (A, B) is controllable and if A is cyclic, then for **almost any** $p \times 1$ vector V, the single-input pair (A, BV) is controllable.

When we say almost any V, we mean that if we select a V matrix **randomly**, there is **virtually probability** 0 that we will end up with one that won't work.

We show intuitively why this is so on an example.

MIMO Cyclic State Feedback Design

Consider the pair $(\mathbf{A}, \mathbf{B})$, with 5 states and 2 inputs, defined by

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 2 \\ 4 & 3 \\ 1 & 0 \end{bmatrix}, \quad \text{and let} \quad \mathbf{BV} = \mathbf{B} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} * \\ * \\ \alpha \\ * \\ \beta \end{bmatrix}.$$

Notice that $\mathbf{A}$ is cyclic, since each of the two distinct eigenvalues 2 and 1, are associated to one and only one Jordan block (respectively of orders 3 and 2).

If we pretend to control $\mathbf{A}$ with the single input built by the product $\mathbf{BV}$, then in order to retain controllability we will need

$$\boxed{\alpha \neq 0} \quad \text{and} \quad \boxed{\beta \neq 0.}$$

MIMO Cyclic State Feedback Design

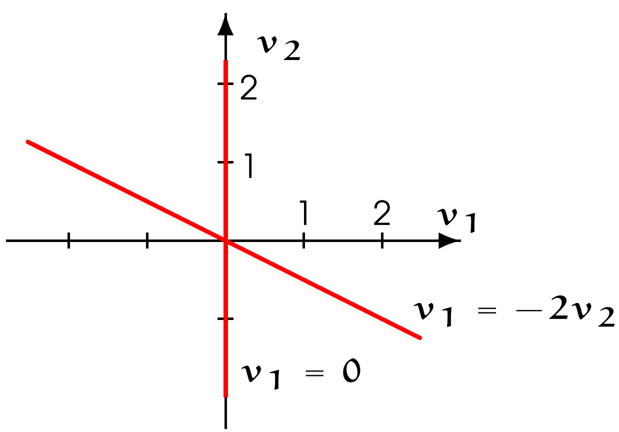
In other words, because

$$\begin{bmatrix} * \\ * \\ \alpha \\ * \\ \beta \end{bmatrix} = \mathbf{BV} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 2 \\ 4 & 3 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} * \\ * \\ v_1 + 2v_2 \\ * \\ v_1 \end{bmatrix},$$

controllability with a single input requires

$$\boxed{v_1 + 2v_2 \neq 0}, \quad \text{and} \quad \boxed{v_1 \neq 0.}$$

which means that in the (v_1, v_2) plane we have to choose a pair (v_1, v_2) **off** the red lines shown in the picture. **Almost any** random pair (v_1, v_2) will satisfy this condition.



MIMO Cyclic State Feedback Design

The condition that A has to be **cyclic** is necessary. For example, for the pair

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}$$

which is controllable, **there is no** $V \in \mathbb{R}^{2\times 1}$ that will make (A, BV) controllable — there are **two** Jordan blocks associated to the same eigenvalue (A is **not** cyclic).

MIMO Cyclic State Feedback Design

The condition that A has to be **cyclic** is necessary. For example, for the pair

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}$$

which is controllable, **there is no** $V \in \mathbb{R}^{2\times 1}$ that will make (A, BV) controllable — there are **two** Jordan blocks associated to the same eigenvalue (A is **not** cyclic).

Can we apply cyclic design to a multi-input controllable pair (A, B) when A is **not cyclic**?

MIMO Cyclic State Feedback Design

The condition that A has to be **cyclic** is necessary. For example, for the pair

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}$$

which is controllable, **there is no** $V \in \mathbb{R}^{2\times 1}$ that will make (A, BV) controllable — there are **two** Jordan blocks associated to the same eigenvalue (A is **not** cyclic).

Can we apply cyclic design to a multi-input controllable pair (A, B) when A is **not cyclic**?

Yes! We just need to **shift the eigenvalues of** A. If we apply a state feedback $u = -K_1 x$, say, to make the eigenvalues of $(A - BK_1)$ all different, we render $(A - BK_1)$ cyclic.

MIMO Cyclic State Feedback Design

Any **randomly chosen** $\mathbf{K}_1$ will generically produce closed-loop eigenvalues that are all different. We can then apply cyclic design to the modified pair $(\mathbf{A} - \mathbf{B}\mathbf{K}_1, \mathbf{B}\mathbf{V})$.

The **Cyclic** design procedure may be summarised as

Procedure for eigenvalue assignment in multi-input state equations by cyclic state feedback design.

1. Check controllability of the n-state, p-input pair $(\mathbf{A}, \mathbf{B})$.
2. Compute a **random** state feedback gain $\mathbf{K}_1 \in \mathbb{R}p \times n$. The matrix $(\mathbf{A} - \mathbf{B}\mathbf{K}_1)$ should be cyclic.
3. Compute a **random** precompensating matrix $\mathbf{V} \in \mathbb{R}^{p \times 1}$. The single input pair $(\mathbf{A} - \mathbf{B}\mathbf{K}_1, \mathbf{B}\mathbf{V})$ should be controllable.
4. Find a state feedback gain $\mathbf{K}_2$ to place the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K}_1 - \mathbf{B}\mathbf{V}\mathbf{K}_2)$ at the desired locations.

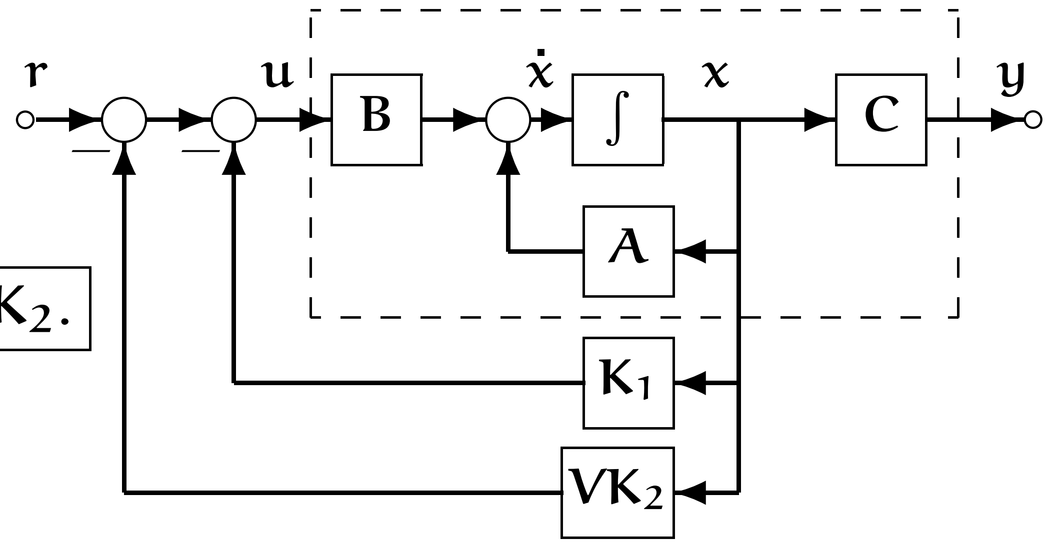
MIMO Cyclic State Feedback Design

In MATLAB, we can use the function `rand` to generate normally distributed random gains

```
[n,p]=size(B);
K1=rand(p,n);
V=rand(p,1);
```

Once we computed $\mathbf{K}_1$, $\mathbf{V}$ and $\mathbf{K}_2$, the total state feedback law is

$$\mathbf{u} = -\mathbf{K}x, \quad \text{with} \quad \boxed{\mathbf{K} = \mathbf{K}_1 + \mathbf{V}\mathbf{K}_2.}$$



Outline

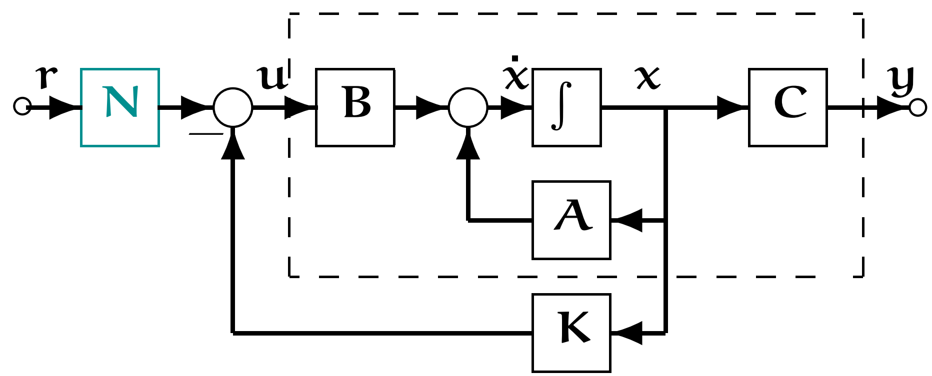
- A Note about Scaling
- MIMO State Feedback Design
  - Cyclic Design
  - MIMO Regulation and Tracking
- MIMO Observers

MIMO Regulation and Tracking

The Regulation and Tracking technique by state feedback is easily extended from SISO to MIMO systems. We just need to be careful with the dimensions of the matrices involved.

MIMO Regulation and Tracking

The Regulation and Tracking technique by state feedback is easily extended from SISO to MIMO systems. We just need to be careful with the dimensions of the matrices involved.



Let's look at the compensation of steady-state tracking error by a feedforward matrix gain $\mathbf{N}$, after a suitable state feedback has been applied to the system. For example, if the plant is **square** (equal number of inputs and outputs), for steady-state tracking we need to satisfy

$$G_K(0)N = C(-A + BK)^{-1}BN = I_{q\times q}$$

$$\Leftrightarrow \quad N = \left[C(-A + BK)^{-1}B\right]^{-1} = G_K(0)^{-1}$$

We need $G_K(0)$ to be **invertible** $\Leftrightarrow$ the plant has no MIMO zeros at $s = 0$.

MIMO Regulation and Tracking

If the plant is **not** square, we distinguish two cases

- **Right invertible plants** with less independent outputs than inputs, $\boxed{\mathbf{q} < \mathbf{p}}$
- **Non-right invertible plants** with more independent outputs than inputs, $\boxed{\mathbf{q} > \mathbf{p}}$

MIMO Regulation and Tracking

If the plant is **not** square, we distinguish two cases

- **Right invertible plants** with less independent outputs than inputs, $\boxed{\mathbf{q} < \mathbf{p}}$
- **Non-right invertible plants** with more independent outputs than inputs, $\boxed{\mathbf{q} > \mathbf{p}}$

Steady-state tracking can be achieved for **all** the outputs **only if** $\mathbf{q} < \mathbf{p}$. If $\mathbf{q} > \mathbf{p}$, we can only achieve steady-state tracking of $\mathbf{p}$ independent outputs (i.e., as many as available control inputs).

MIMO Regulation and Tracking

If the plant is **not** square, we distinguish two cases

- **Right invertible plants** with less independent outputs than inputs, $\boxed{q < p}$
- **Non-right invertible plants** with more independent outputs than inputs, $\boxed{q > p}$

Steady-state tracking can be achieved for **all** the outputs **only if** $q < p$. If $q > p$, we can only achieve steady-state tracking of p independent outputs (i.e., as many as available control inputs).

Example (Tracking in Right Invertible Plants). Suppose that, after state-feedback in a 2×3 plant, we have achieved

$$G_K(s) = \begin{bmatrix} \frac{3}{s+1} & \frac{10s+1}{s^2+2s+1} & 0 \\ 0 & \frac{s+5}{s^2+2s+1} & 0 \end{bmatrix}$$

MIMO Regulation and Tracking

Example (Continuation). We wish to obtain a feedforward gain $\mathbf{N}$ such that

$$\mathbf{G_K}(0)\mathbf{N} = \mathbf{I}_{2\times 2}$$

Because $\mathbf{G_K}(s)$ is "short and wide", and has full row rank at $s = 0$,

$$\mathbf{G_K}(0) = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 5 & 0 \end{bmatrix},$$

we can select $\mathbf{N}$ as

$$\mathbf{G_K}(0)\mathbf{N} = \mathbf{G_K}(0) \underbrace{\mathbf{G_K}(0)^T \left[\mathbf{G_K}(0)\mathbf{G_K}(0)^T\right]^{-1}}_{\mathbf{N}} = \mathbf{I}_{2\times 2}$$

The solution is then

$$\mathbf{N} = \begin{bmatrix} 1/3 & -1/15 \\ 0 & 1/5 \\ 0 & 0 \end{bmatrix} \qquad \square$$

MIMO Regulation and Tracking

Example (Tracking in Non-Right Invertible Plants). Suppose now that, after state-feedback in a 3×2 plant, we have achieved

$$\mathbf{G_K}(s) = \begin{bmatrix} \frac{3}{s+1} & 0 \\ \frac{10s+1}{s^2+2s+1} & \frac{s+5}{s^2+2s+1} \\ 0 & 0 \end{bmatrix}$$

At $s = 0$ we have $\mathbf{G_K}(0) = \begin{bmatrix} 3 & 0 \\ 10 & 5 \\ 0 & 0 \end{bmatrix}$ which is **not** full row rank (there are 3 independent rows, but the rank of the matrix is only 2. Hence, it is impossible to find $\mathbf{N} \in \mathbb{R}^{2\times 3}$ such that

$$\mathbf{G_K}(0)\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□

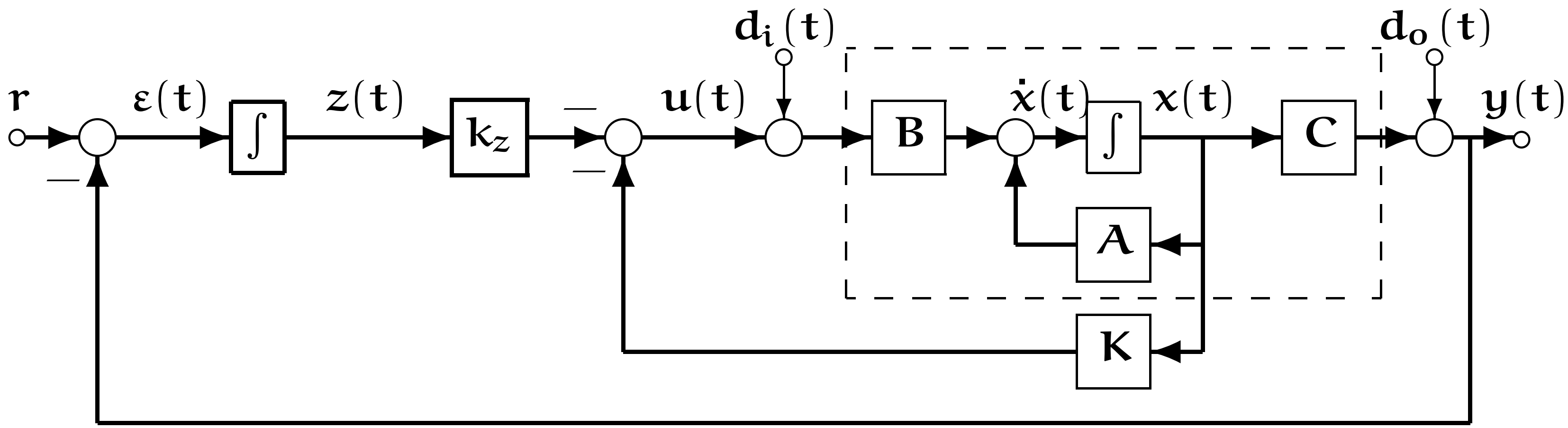
MIMO Regulation and Tracking

Tracking with **Integral Action** is subject to the same restrictions: **we can only achieve asymptotic tracking of a maximum of as many outputs as control inputs are available.**

MIMO Regulation and Tracking

Tracking with **Integral Action** is subject to the same restrictions: **we can only achieve asymptotic tracking of a maximum of as many outputs as control inputs are available.**

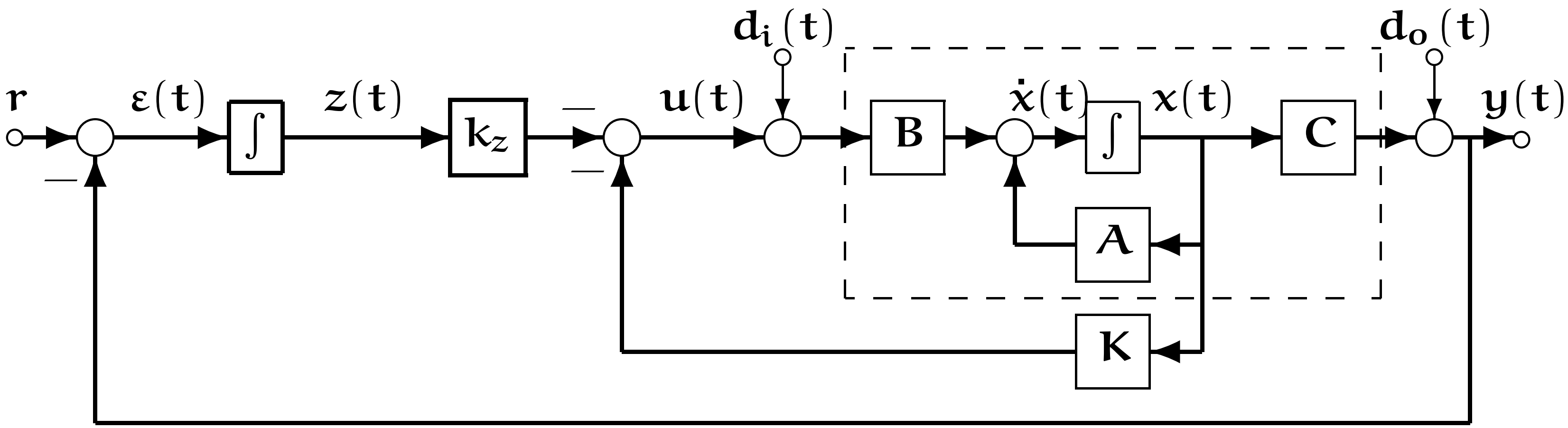
The scheme and computation procedure is the same as in SISO



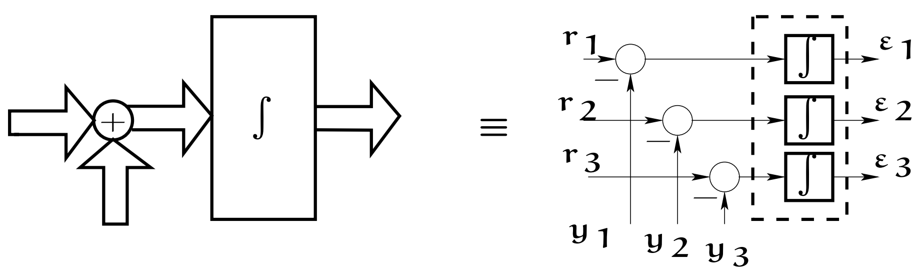
MIMO Regulation and Tracking

Tracking with **Integral Action** is subject to the same restrictions: **we can only achieve asymptotic tracking of a maximum of as many outputs as control inputs are available.**

The scheme and computation procedure is the same as in SISO



Note that now the **integral action** is applied to each of the q reference input channels.



MIMO Regulation and Tracking

The procedure to compute $\mathbf{K}$ and $\mathbf{k}_z$ for the state feedback control with integral action is exactly as in the SISO case,

$$\dot{z}(t) = \mathbf{r} - \mathbf{y}(t) = \mathbf{r} - \mathbf{C}\mathbf{x}(t)$$

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{K} & \mathbf{k}_z \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ z(t) \end{bmatrix}$$

where $\mathbf{K}_a = [\,\mathbf{K}\ \mathbf{k}_z\,]$ is computed to place the eigenvalues of the **augmented plant** $(\mathbf{A}_a, \mathbf{B}_a)$ at desired locations, where

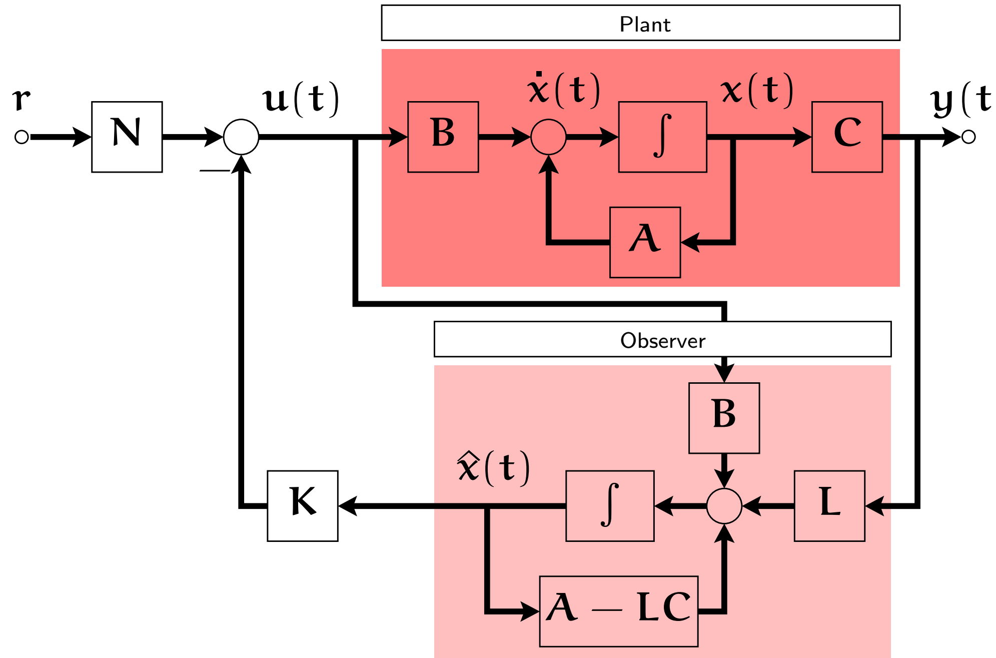
$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & 0_{n\times q} \\ -\mathbf{C} & 0_{q\times q} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{B} \\ 0_{q\times p} \end{bmatrix}$$

Outline

- A Note about Scaling
- MIMO State Feedback Design
  - Cyclic Design
  - MIMO Regulation and Tracking
- MIMO Observers

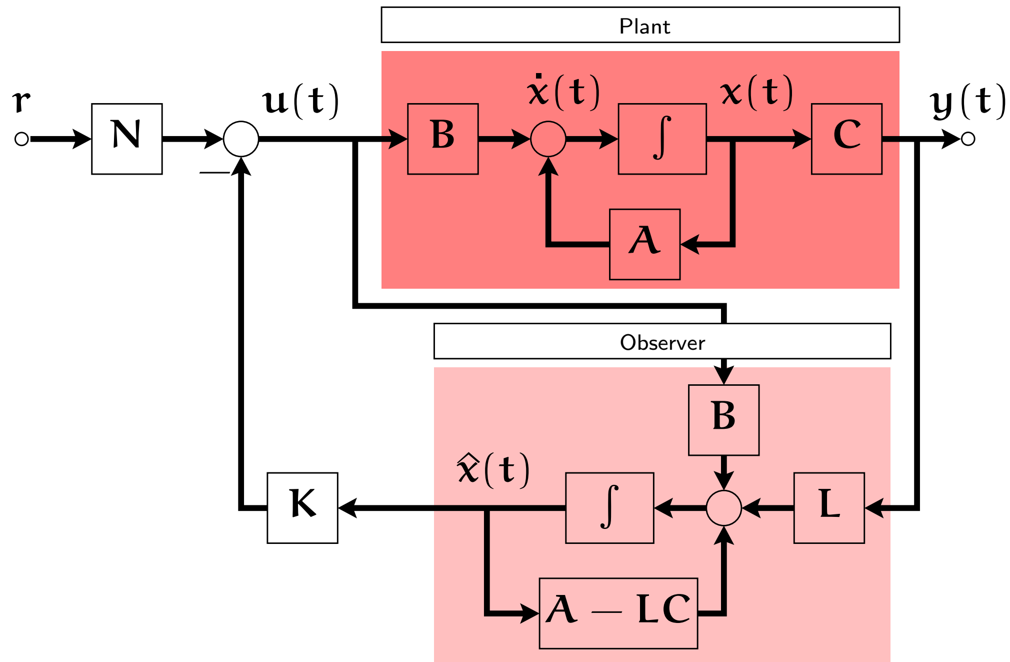
MIMO Observers

Design of MIMO obervers extends directly from the SISO case. To compute the observer gain $\mathbf{L}$ we can use "duality" and the procedures seen to compute state feedback gains, as the **cyclic design**.



MIMO Observers

Design of MIMO obervers extends directly from the SISO case. To compute the observer gain $\mathbf{L}$ we can use "duality" and the procedures seen to compute state feedback gains, as the **cyclic design**.



Alternatively, we can also use **linear quadratic optimal design**, to obtain an optimal observer gain $\mathbf{L}$. The observer obtained in this way is usually called the **Kalman filter**.

Reduced Order Observers

- A technique that could be particularly useful in the MIMO case is that of **reduced order observer** design.

Reduced Order Observers

- A technique that could be particularly useful in the MIMO case is that of **reduced order observer** design.
- One argument against state-feedback+observer controllers is that they generally have high order:

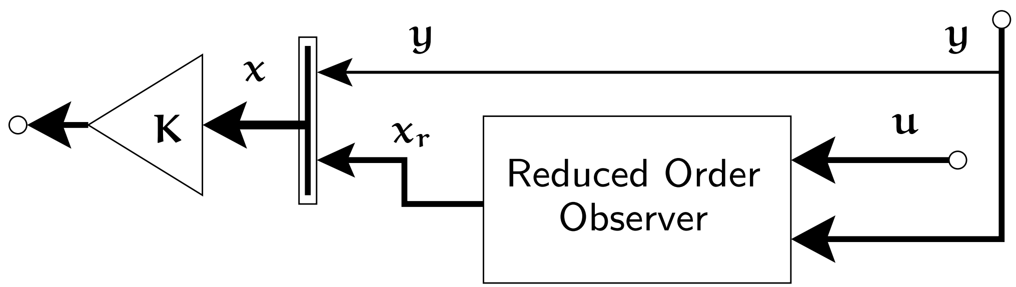
  since the observer includes a **model of the plant**, we would normally obtain a controller of at least of the same order of the open-loop system.

Reduced Order Observers

However, it is possible to **reduce** this order to some extent by designing a **reduced order observer**.

The key observation is that

if the state equations are selected in such a way that the q system outputs constitute the first q states, then in fact we only need to estimate the remaining $n - q$ states to apply state feedback.



Reduced Order Observers

Let the system equations be

$$\begin{bmatrix} \dot{y} \\ \dot{x}_r \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x_r \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

where the first q states are the outputs of the system. Because we **measure** the outputs, we could try and build an observer **only** to estimate the remaining states x_r.

$$\dot{x}_r = A_{22}x_r + A_{21}y + B_2u$$
$$y_r \triangleq A_{12}x_r$$

Reduced Order Observers

Let the system equations be

$$\begin{bmatrix} \dot{y} \\ \dot{x}_r \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} y \\ x_r \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

where the first q states are the outputs of the system. Because we **measure** the outputs, we could try and build an observer **only** to estimate the remaining states x_r.

$$\begin{aligned} \dot{x}_r &= A_{22}x_r + A_{21}y + B_2u \\ y_r &\triangleq A_{12}x_r = \underbrace{\dot{y} - A_{11}y - B_1u}_{\text{measurable "output"}} \end{aligned}$$

We can think of y_r as a "virtual" output of a reduced order state equation with state x_r. We can compute everything in y_r from measurements; the only problem is that it appears we need $\dot{y}$.

Reduced Order Observers

Then the observer required to estimate the states $\mathbf{x_r}$ can be constructed as

$$\dot{\hat{\mathbf{x}}}_{\mathbf{r}} = \mathbf{A}_{22}\hat{\mathbf{x}}_{\mathbf{r}} + \mathbf{A}_{21}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{L}_{\mathbf{r}}\left(\mathbf{y}_{\mathbf{r}} - \mathbf{A}_{12}\hat{\mathbf{x}}_{\mathbf{r}}\right)$$

Reduced Order Observers

Then the observer required to estimate the states $\mathbf{x_r}$ can be constructed as

$$\begin{aligned}\dot{\hat{\mathbf{x}}}_\mathbf{r} &= \mathbf{A}_{22}\hat{\mathbf{x}}_\mathbf{r} + \mathbf{A}_{21}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{L}_\mathbf{r}\left(\mathbf{y}_\mathbf{r} - \mathbf{A}_{12}\hat{\mathbf{x}}_\mathbf{r}\right) \\ &= (\mathbf{A}_{22} - \mathbf{L}_\mathbf{r}\mathbf{A}_{12})\hat{\mathbf{x}}_\mathbf{r} + \mathbf{A}_{21}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{L}_\mathbf{r}\mathbf{y}_\mathbf{r}\end{aligned}$$

By designing the observer gain $\mathbf{L}_\mathbf{r}$ to make $(\mathbf{A}_{22} - \mathbf{L}_\mathbf{r}\mathbf{A}_{12})$ Hurwitz, we guarantee asymptotic convergence of the estimates $\hat{\mathbf{x}}_\mathbf{r}$ to $\mathbf{x}_\mathbf{r}$.

Reduced Order Observers

Then the observer required to estimate the states x_r can be constructed as

$$\begin{aligned}\dot{\hat{x}}_r &= A_{22}\hat{x}_r + A_{21}y + B_2u + L_r\left(y_r - A_{12}\hat{x}_r\right)\\ &= (A_{22} - L_rA_{12})\hat{x}_r + A_{21}y + B_2u + L_ry_r\end{aligned}$$

By designing the observer gain L_r to make $(A_{22} - L_rA_{12})$ Hurwitz, we guarantee asymptotic convergence of the estimates $\hat{x}_r$ to x_r.

The reduced order observer is a system of order $r = n - q$, in contrast with a full observer, which is of order n.

Reduced Order Observers

Then the observer required to estimate the states $\mathbf{x_r}$ can be constructed as

$$\begin{aligned}\dot{\hat{\mathbf{x}}}_r &= \mathbf{A}_{22}\hat{\mathbf{x}}_r + \mathbf{A}_{21}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{L}_r\left(\mathbf{y}_r - \mathbf{A}_{12}\hat{\mathbf{x}}_r\right)\\ &= (\mathbf{A}_{22} - \mathbf{L}_r\mathbf{A}_{12})\hat{\mathbf{x}}_r + \mathbf{A}_{21}\mathbf{y} + \mathbf{B}_2\mathbf{u} + \mathbf{L}_r\mathbf{y}_r\end{aligned}$$

By designing the observer gain $\mathbf{L}_r$ to make $(\mathbf{A}_{22} - \mathbf{L}_r\mathbf{A}_{12})$ Hurwitz, we guarantee asymptotic convergence of the estimates $\hat{\mathbf{x}}_r$ to $\mathbf{x}_r$.

The reduced order observer is a system of order $r = n - q$, in contrast with a full observer, which is of order n.

The only “problem” with the reduced observer is that it appears that we need to **differentiate** the output to compute $\mathbf{y}_r$!

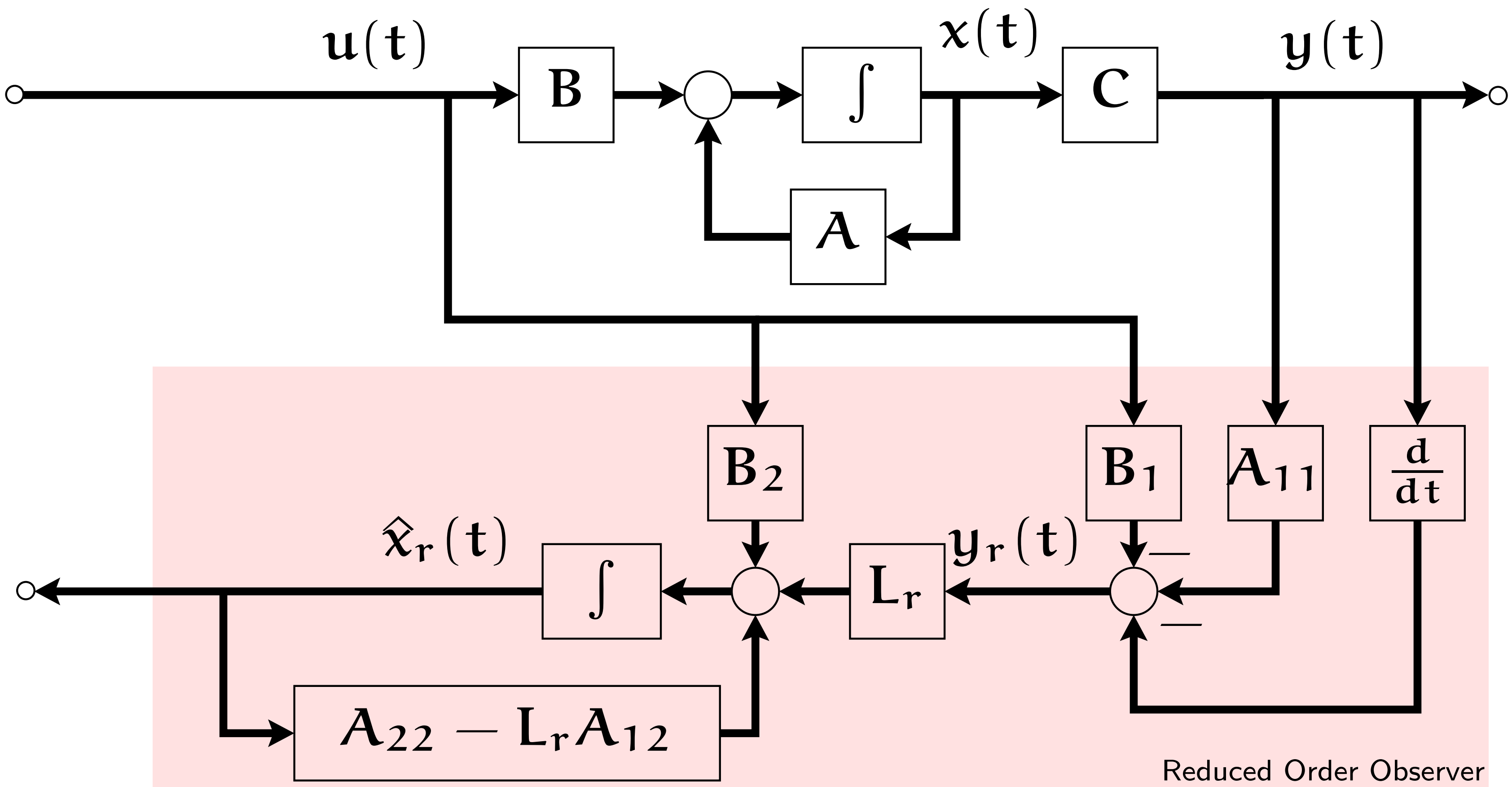
$$\mathbf{y}_r = \dot{\mathbf{y}} - \mathbf{A}_{11}\mathbf{y} - \mathbf{B}_1\mathbf{u}$$

Reduced Order Observers

We eliminate the need to differentiate the output by clever implementation. We show how by block diagram algebra.

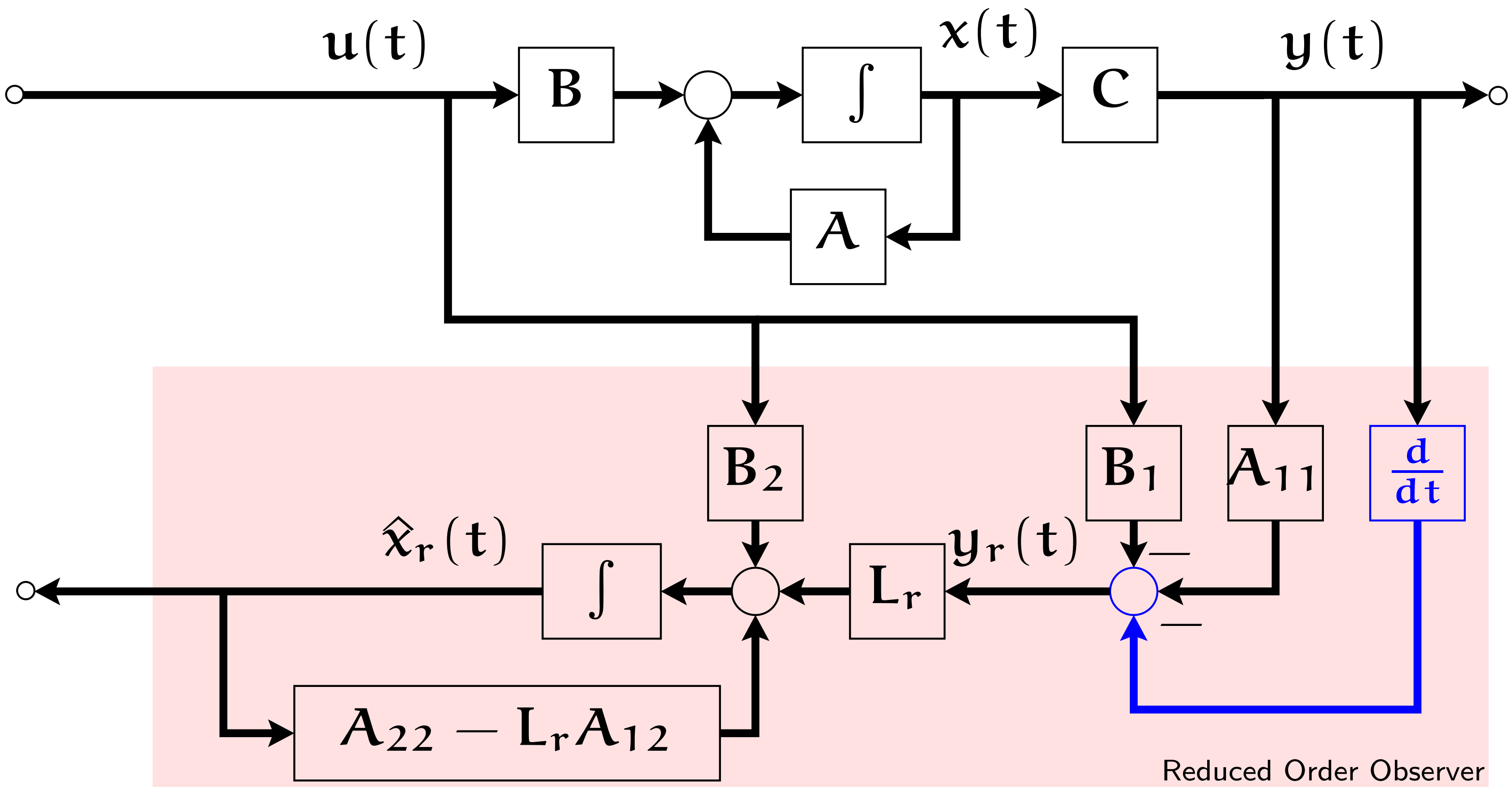
Reduced Order Observers

We eliminate the need to differentiate the output by clever implementation. We show how by block diagram algebra.



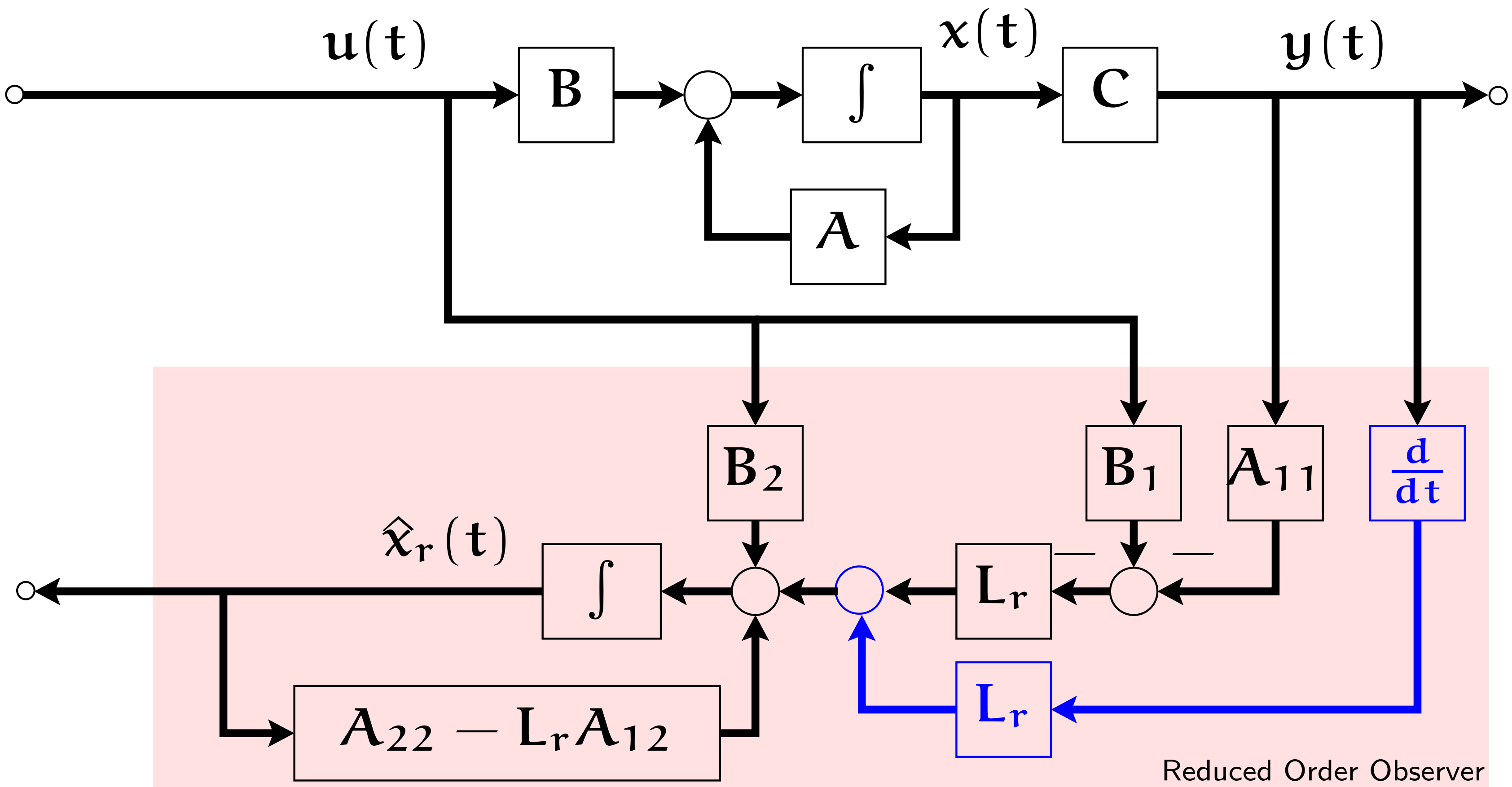
Reduced Order Observers

We eliminate the need to differentiate the output by clever implementation. We show how by block diagram algebra.



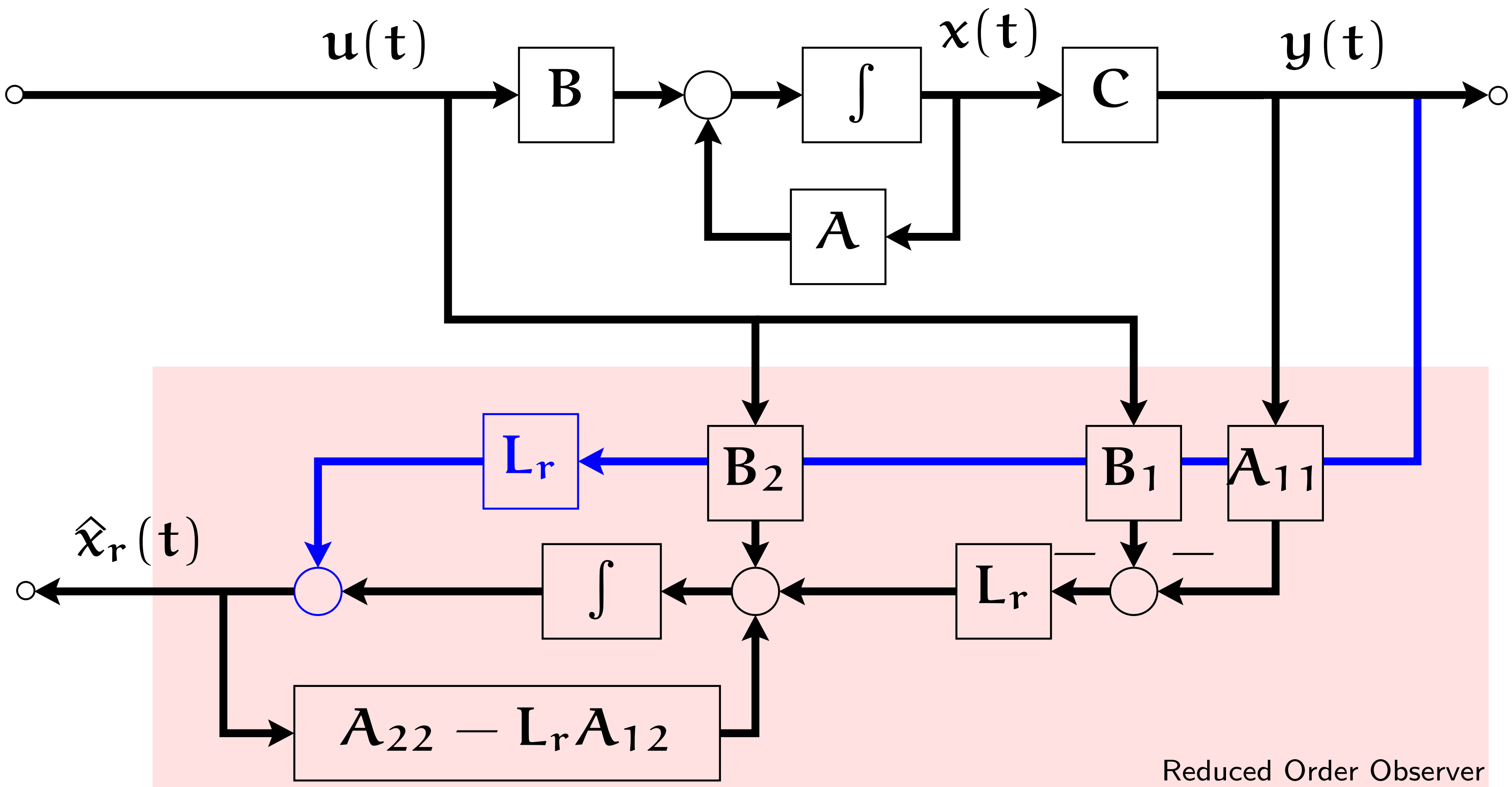
Reduced Order Observers

We eliminate the need to differentiate the output by clever implementation. We show how by block diagram algebra.



Reduced Order Observers

We eliminate the need to differentiate the output by clever implementation. We show how by block diagram algebra.



Summary

- **Scaling:** The important points to remember:

Summary

- **Scaling:** The important points to remember:
  - State-space models are, in general, the most reliable models for subsequent computations.

Summary

- **Scaling:** The important points to remember:
  - State-space models are, in general, the most reliable models for subsequent computations.
  - Scaling model data can improve the accuracy of your results.

Summary

- **Scaling:** The important points to remember:
  - State-space models are, in general, the most reliable models for subsequent computations.
  - Scaling model data can improve the accuracy of your results.
  - Numerical computing is a tricky business, and virtually all computer tools can fail under certain conditions.

Summary

- **Scaling:** The important points to remember:
  - State-space models are, in general, the most reliable models for subsequent computations.
  - Scaling model data can improve the accuracy of your results.
  - Numerical computing is a tricky business, and virtually all computer tools can fail under certain conditions.
- **MIMO state feedback and observer design** extends directly from SISO to MIMO systems. One significant difference in state feedback for MIMO systems:

Summary

- **Scaling:** The important points to remember:
  - State-space models are, in general, the most reliable models for subsequent computations.
  - Scaling model data can improve the accuracy of your results.
  - Numerical computing is a tricky business, and virtually all computer tools can fail under certain conditions.
- **MIMO state feedback and observer design** extends directly from SISO to MIMO systems. One significant difference in state feedback for MIMO systems:
  - the state feedback gain $\mathbf{K}$ that place the closed-loop eigenvalues of a MIMO system at desired locations is **not unique**

Summary

- The "excess" of degrees of freedom in MIMO state feedback design can be handled by

Summary

- The "excess" of degrees of freedom in MIMO state feedback design can be handled by
  - **Cyclic design**, which reduces the problem to one of SIMO — it requires the A matrix to be **cyclic**, or otherwise, an **extra** state feedback to render it cyclic.

Summary

- The "excess" of degrees of freedom in MIMO state feedback design can be handled by
  - **Cyclic design**, which reduces the problem to one of SIMO — it requires the A matrix to be **cyclic**, or otherwise, an **extra** state feedback to render it cyclic.
  - **Optimal design**, for example LQR, as we will see in coming lectures.

Summary

- The "excess" of degrees of freedom in MIMO state feedback design can be handled by
  - **Cyclic design**, which reduces the problem to one of SIMO — it requires the A matrix to be **cyclic**, or otherwise, an **extra** state feedback to render it cyclic.
  - **Optimal design**, for example LQR, as we will see in coming lectures.
- MIMO regulation and tracking can be handled by the same procedures used for SISO systems. Yet, one must be careful that the tracking objectives be consistent with the **feasible tracking directions** that the plant can reach.

Summary

- **MIMO observers** are designed in the same way as SISO observers, and the state feedback design techniques can be used via **duality**.

Summary

- **MIMO observers** are designed in the same way as SISO observers, and the state feedback design techniques can be used via **duality**.
- Particularly interesting in the MIMO case is the possibility of designing **reduced order observers**, which can significantly reduce the order of the overall state feedback + observer controller.