Discrete-time State feedback

Kjartan Halvorsen

November 23, 2021

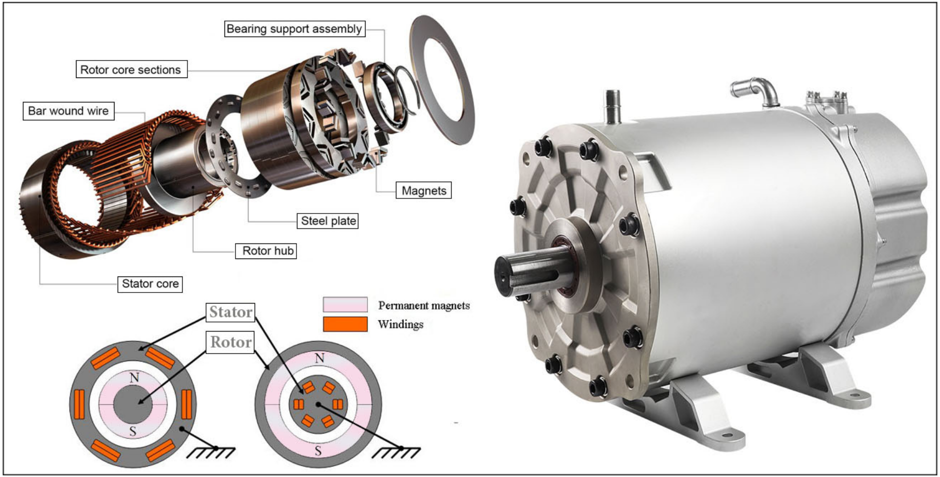
The discrete-time state-space model

The discrete-time state-space model

$$\xrightarrow{u(k)} \boxed{\begin{array}{c} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) \end{array}} \xrightarrow{y(k)}$$

Obtain state-space model from discrete-time pulse-transfer function

The permanent magnet synchronous motor



Permanent Magnet Synchronous Motor Construction

The PMSM

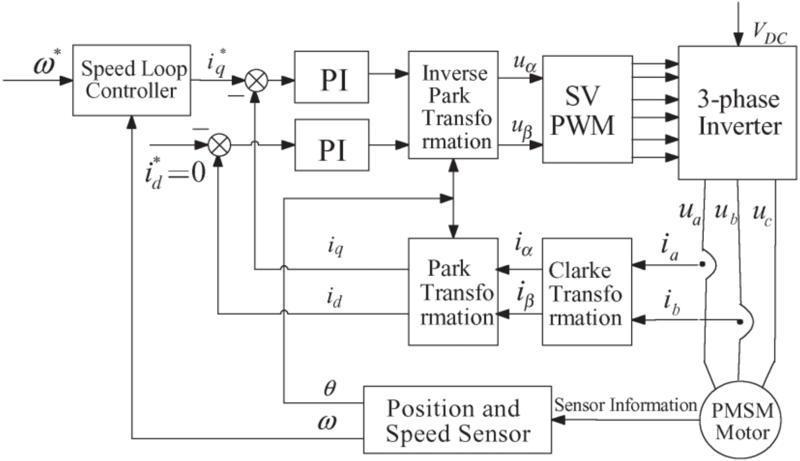


Fig. 1. Block diagram of the PMSM control system.

Liu and Li "Speed control for PMSM servo system", IEEE Transactions on Industrial Electronics, 2012.

Identified model

Two poles, two zeros, one delay

$$u(k) \longrightarrow \boxed{\frac{1}{z}} \longrightarrow \boxed{\frac{b_0 z^2+b_1 z+b_2}{z^2+a_1 z+a_2}} \longrightarrow y(k)$$

$$\Updownarrow$$

$$u(k) \longrightarrow \boxed{\frac{b_0 z^2+b_1 z+b_2}{z^2+a_1 z+a_2}} \longrightarrow \boxed{\frac{1}{z}} \longrightarrow y(k)$$

$$\Updownarrow$$

$$u(k) \longrightarrow \boxed{\frac{b_0 z^2+b_1 z+b_2}{z(z^2+a_1 z+a_2}\,)} \longrightarrow y(k)$$

Identified model

$$H(z) = \frac{6.91z^2 + 16.48z - 17.87}{z(z^2 - 1.766z + 0.7665)} = \frac{6.91(z + 3.19)(z - 0.81)}{z(z - 0.998)(z - 0.768)}$$

From pulse-transfer function to state space model

$$u(k) \longrightarrow \boxed{H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z(z^2 + a_1 z + a_2)}} \longrightarrow y(k)$$

$$\Updownarrow$$

$$u(k) \longrightarrow \boxed{\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= C x(k) \end{aligned}} \longrightarrow y(k)$$

Canonical forms

Given pulse-transfer function

$$H(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}.$$

Find a representation in state space form

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= C x(k) \end{aligned}$$

Canonical forms

Given pulse-transfer function

$$H(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}.$$

Find a representation in state space form

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= C x(k) \end{aligned}$$

- Controlable canonical form
- Observable canonical form

Controllable canonical form

Given pulse-transfer function

$$H(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}.$$

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} x(k) \end{aligned}$$

Observable canonical form

Given pulse-transfer function

$$H(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}.$$

$$\begin{aligned} x(k+1) &= \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(k) \end{aligned}$$

Canonical forms

Activity Find the controllable canonical form for the pulse-transfer function of the motor (needed for question 2 on the exercises).

$$H(z) = \frac{6.91z^2 + 16.48z - 17.87}{z(z^2 - 1.766z + 0.7665)} = \frac{6.91(z + 3.19)(z - 0.81)}{z(z - 0.998)(z - 0.768)}$$

Canonical forms

Activity Find the controllable canonical form for the pulse-transfer function of the motor (needed for question 2 on the exercises).

$$H(z) = \frac{6.91z^2 + 16.48z - 17.87}{z(z^2 - 1.766z + 0.7665)} = \frac{6.91(z + 3.19)(z - 0.81)}{z(z - 0.998)(z - 0.768)}$$

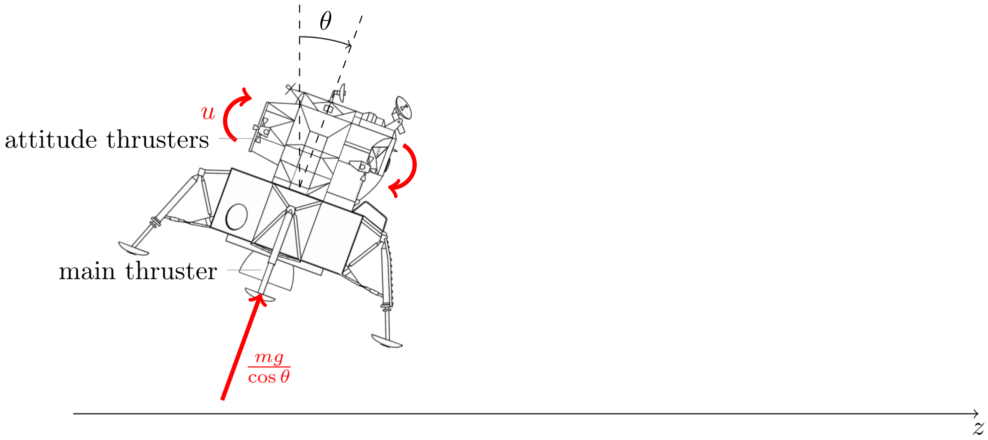
Given pulse-transfer function

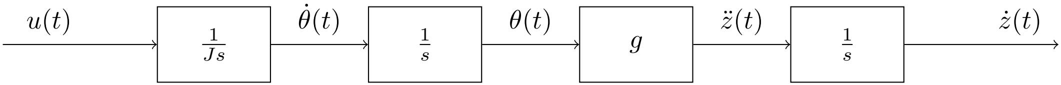
$$H(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}.$$

$$x(k+1) = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} x(k)$$

Discretizing a continuous-time state-space model

Example - The Apollo lunar module





Example - The Apollo lunar module

State variables: $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} \dot{\theta} & \theta & \dot{z} \end{bmatrix}^T$. With dynamics

$$\begin{cases} \dot{x}_1 = \ddot{\theta} = \frac{1}{J} u \\ \dot{x}_2 = \dot{\theta} = x_1 \\ \dot{x}_3 = \ddot{z} = g\theta = g x_2 \end{cases}$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{J} \\ 0 \\ 0 \end{bmatrix}}_{B} u$$

Discretization

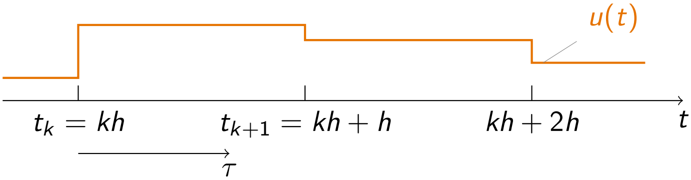
The general solution to a linear, continuous-time state-space system

$$x(t_k + \tau) = \mathrm{e}^{A(\tau)} x(t_k) + \int_0^{\tau} \mathrm{e}^{As} Bu\big((t_k + \tau) - s\big) ds$$

Discretization

The general solution to a linear, continuous-time state-space system

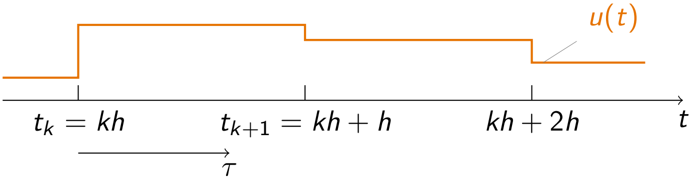
$$x(t_k + \tau) = \mathrm{e}^{A(\tau)} x(t_k) + \int_0^{\tau} \mathrm{e}^{As} B u\big((t_k + \tau) - s\big) ds$$



Discretization

The general solution to a linear, continuous-time state-space system

$$x(t_k+\tau) = \mathrm{e}^{A(\tau)}x(t_k) + \int_0^{\tau} \mathrm{e}^{As}Bu\big((t_k+\tau)-s\big)ds$$



$$\begin{aligned} x(kh+h) &= \mathrm{e}^{Ah}x(kh) + \int_0^{h} \mathrm{e}^{As}Bu(kh+h-s)ds \\ &= \underbrace{\mathrm{e}^{Ah}}_{\Phi(h)}x(kh) + \underbrace{\left(\int_0^{h} \mathrm{e}^{As}Bds\right)}_{\Gamma(h)}u(kh) \end{aligned}$$

Discretization - The matrix exponential

SIAM REVIEW
Vol. 45, No. 1, pp. 3–49

Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later*

Cleve Moler†
Charles Van Loan‡

Abstract. In principle, the exponential of a matrix could be computed in many ways. Methods involving approximation theory, differential equations, the matrix eigenvalues, and the matrix characteristic polynomial have been proposed. In practice, consideration of computational stability and efficiency indicates that some of the methods are preferable to others but that none are completely satisfactory.

Most of this paper was originally published in 1978. An update, with a separate bibliography, describes a few recent developments.

Discretization - The matrix exponential

Square matrix A. Scalar variable t.

$$\mathrm{e}^{At} = I + At + \frac{t^2}{2!}A^2 + \frac{t^3}{3!}A^3 + \cdots$$

Laplace transform

$$\mathcal{L}\left\{\mathrm{e}^{At}\right\} = (sI - A)^{-1}$$

Discretization - example

$$x(kh+h) = e^{Ah}x(kh) + \int_0^h e^{As}Bu(kh+h-s)ds$$
$$= \underbrace{e^{Ah}}_{\Phi(h)} x(kh) + \underbrace{\left(\int_0^h e^{As}Bds\right)}_{\Gamma(h)} u(kh)$$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ g & 0 & 0 \end{bmatrix}, \quad A^3 = 0$$

So,

$$\Phi(h) = e^{Ah} = I + Ah + A^2h^2/2 + \cdots$$

Discretization - example

$$x(kh+h) = e^{Ah}x(kh) + \int_0^h e^{As}Bu(kh+h-s)ds$$
$$= \underbrace{e^{Ah}}_{\Phi(h)} x(kh) + \underbrace{\left(\int_0^h e^{As}Bds\right)}_{\Gamma(h)} u(kh)$$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix}, \quad A^2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ g & 0 & 0 \end{bmatrix}, \quad A^3 = 0$$

So,

$$\Phi(h) = e^{Ah} = I + Ah + A^2h^2/2 + \cdots$$

$$\Phi(h) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & g & 0 \end{bmatrix} h + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ g & 0 & 0 \end{bmatrix} \frac{h^2}{2} = \begin{bmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ \frac{h^2 g}{2} & hg & 1 \end{bmatrix}$$

Discretization - example

$$
\begin{aligned}
x(kh+h) &= e^{Ah}x(kh) + \int_0^h e^{As}Bu(kh+h-s)ds \\
&= \underbrace{e^{Ah}}_{\Phi(h)}x(kh) + \underbrace{\left(\int_0^h e^{As}Bds\right)}_{\Gamma(h)}u(kh)
\end{aligned}
$$

$$
e^{As}B = \begin{bmatrix} 1 & 0 & 0 \\ s & 1 & 0 \\ \frac{s^2g}{2} & sg & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{J} \\ 0 \\ 0 \end{bmatrix} = \frac{1}{J}\begin{bmatrix} 1 \\ s \\ \frac{gs^2}{2} \end{bmatrix}
$$

$$
\Gamma(h) = \int_0^h e^{As}Bds = \frac{1}{J}\int_0^h \begin{bmatrix} 1 \\ s \\ \frac{gs^2}{2} \end{bmatrix} ds = \frac{1}{J}\begin{bmatrix} h \\ \frac{h^2}{2} \\ \frac{gh^3}{6} \end{bmatrix}
$$

Discretization - example

$$
\begin{aligned}
x(kh+h) &= \mathrm{e}^{Ah}x(kh) + \int_0^h \mathrm{e}^{As}Bu(kh+h-s)ds \\
&= \underbrace{\mathrm{e}^{Ah}}_{\Phi(h)}x(kh) + \underbrace{\left(\int_0^h \mathrm{e}^{As}Bds\right)}_{\Gamma(h)}u(kh) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ \frac{h^2 g}{2} & hg & 1 \end{bmatrix} x(kh) + \frac{1}{J}\begin{bmatrix} h \\ \frac{h^2}{2} \\ \frac{gh^3}{6} \end{bmatrix} u(kh)
\end{aligned}
$$

Stability

Eigenvalues and eigenvectors

Definition The eigenvalues $\lambda_i \in \mathbb{R}$ and eigenvectors $v_i \in \mathbb{R}^n$ of a matrix $\Phi \in \mathbb{R}^{n\times n}$ are the n pairs $(\lambda_i, v_i \neq 0),\ i = 1, 2, \ldots, n$ that satisfy

$$\Phi v_i = \lambda_i v_i$$

Stability

The system

$$x(k+1) = \Phi x(k), \quad x(0) = x_0$$

is stable if $\lim_{t \to \infty} x(kh) = 0, \quad \forall\, x_0 \in \mathbb{R}^n$.

A necessary and sufficient requirement for stability is that all the eigenvalues of Φ are inside the unit circle.

The eigenvalues of Φ are the poles of the system.

State feedback control

State feedback control

Given

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{1}$$

and measurements (or an estimate) of the state vector $x(k)$.

Linear state feedback is the control law

$$\begin{aligned} u(k) = f\big((x(k), u_c(k)\big) &= -l_1 x_1(k) - l_2 x_2(k) - \cdots - l_n x_n(k) + l_0 u_c(k) \\ &= -Lx(k) + l_0 u_c(k), \end{aligned}$$

where

$$L = \begin{bmatrix} l_1 & l_2 & \cdots & l_n \end{bmatrix}.$$

Substituting this in the state-space model (1) gives

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2) \cdots (z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Linear state feedback gives the system

$$x(k+1) = (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \tag{4}$$

with characteristic polynomial

$$\det\left(zI - (\Phi - \Gamma L)\right) = z^n + \beta_1(l_1, \ldots, l_n) z^{n-1} + \cdots \beta_n(l_1, \ldots, l_n). \tag{5}$$

Pole placement by state feedback

Given (or choosing) a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2) \cdots (z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n. \tag{3}$$

Linear state feedback gives the system

$$x(k+1) = (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \tag{4}$$

with characteristic polynomial

$$\det\left(zI - (\Phi - \Gamma L)\right) = z^n + \beta_1(l_1, \ldots, l_n) z^{n-1} + \cdots \beta_n(l_1, \ldots, l_n). \tag{5}$$

Set the coefficients of the desired characteristic polynomial (30) equal to the coefficients of (5) to obtain the system of equations

$$\begin{aligned}
\beta_1(l_1, \ldots, l_n) &= \alpha_1 \\
\beta_2(l_1, \ldots, l_n) &= \alpha_2 \\
&\vdots \\
\beta_n(l_1, \ldots, l_n) &= \alpha_n
\end{aligned}$$

Pole placement by state feedback

The system of equations

$$
\begin{aligned}
\beta_1(l_1, \ldots, l_n) &= \alpha_1 \\
\beta_2(l_1, \ldots, l_n) &= \alpha_2 \\
&\vdots \\
\beta_n(l_1, \ldots, l_n) &= \alpha_n
\end{aligned}
$$

is always linear in the parameters of the controller, hence

$$
ML^{\mathrm{T}} = \alpha,
$$

where $\alpha^{\mathrm{T}} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix}$.

Pole placement by state feedback

Given a desired placement of the closed-loop poles $p_1, p_2, \ldots, p_n$, being roots of the desired characteristic polynomial

$$a_c(z) = (z - p_1)(z - p_2)\cdots(z - p_n) = z^n + \alpha_1 z^{n-1} + \cdots \alpha_n.$$

and closed-loop system

$$\begin{aligned} x(k+1) &= (\Phi - \Gamma L)\, x(k) + l_0 \Gamma u_c(k) \\ y(k) &= Cx(k) \end{aligned}$$

The Matlab (*control systems toolbox*) has methods for computing the gain vector L

1. Ackerman's method

```
L = acker(Phi, Gamma, pd)
```

2. Numerically more stable method

```
L = place(Phi, Gamma, pd)
```

The reference input gain l_0

The closed-loop state space system

$$
\begin{aligned}
x(k+1) &= \underbrace{(\Phi - \Gamma L)}_{\Phi_c} x(k) + l_0 \Gamma u_c(k) \\
y(k) &= Cx(k)
\end{aligned}
$$

with constant reference signal $u_c(k) = u_{c,f}$ has the steady-state solution $(x(k+1) = x(k))$

The reference input gain l_0

The closed-loop state space system

$$x(k+1) = \underbrace{(\Phi - \Gamma L)}_{\Phi_c} x(k) + l_0 \Gamma u_c(k)$$

$$y(k) = Cx(k)$$

with constant reference signal $u_c(k) = u_{c,f}$ has the steady-state solution $(x(k+1) = x(k))$

$$x_f = l_0 (I - \Phi_c)^{-1} \Gamma u_{c,f}$$

$$y_f = Cx_f = l_0 C (I - \Phi_c)^{-1} \Gamma u_{c,f}.$$

We want $y_f = u_{c,f}$,

$$\Rightarrow \qquad l_0 = \frac{1}{C(I - \Phi_c)^{-1}\Gamma}$$