

# Contents

<b>1 gain scheduling</b>	<b>1</b>
1.1 syms matlab, linearization . . . . .	1
1.1.1 Video Lecture . . . . .	1
1.1.2 Definitions . . . . .	1
1.2 full state feedback . . . . .	2
1.2.1 Definitions . . . . .	2
1.3 Iterative Eigenvalue Manipulation . . . . .	2
<b>2 General Tools</b>	<b>2</b>
2.1 MATLAB . . . . .	2

## 1 gain scheduling

### 1.1 syms matlab, linearization

#### 1.1.1 Video Lecture

I link below a short (and dense) example of the Linearization of a nonlinear set of dynamics using both MATLAB syms and a manual method:

Linerization Example with syms in MATLAB

#### 1.1.2 Definitions

We are some system described by the following equation, for state  $x$  and input  $u$ :

$$\dot{x} = f(x, u)$$

, where our linearization is defined, about a linearization point  $(x_0, u_0)$ :

$$\dot{x}(x, u) \approx f(x_0, u_0) + \left. \frac{\partial f}{\partial x} \right|_{x_0, u_0} (x - x_0) + \left. \frac{\partial f}{\partial u} \right|_{x_0, u_0} (u - u_0)$$

The jacobians are redefined to be  $A(x_0, u_0)$  and  $B(x_0, u_0)$  in order of appearance. We can then group terms and define  $(\Delta x, \Delta u) := (x - x_0, u - u_0)$ , such that:

$$\Delta \dot{x} = A \Delta x + B \Delta u$$

Note the linearity of the derivative, such that  $\frac{d \Delta x}{dt} = \Delta \frac{dx}{dt}$ .

## 1.2 full state feedback

### 1.2.1 Definitions

I will assume you have seen or understand the prior definitions for the linearization of our nonlinear dynamics. Given them, We define full state feedback as such:

$$u = -K^T x \implies \Delta x = (A - BK^T)\Delta x$$

It should be noted  $u_0 = -K^T x_0 \implies \Delta u = -K^T \Delta x$

The matrix  $A - BK^T$ 's eigenvalues decide directly the settling time and stability margins for our linear system. Because our system is heavily nonlinear and MIMO (Multiple Input, Multiple Output), we cannot simply solve for the eigenvalues and remain computationally efficient. Instead, can use `place()`, in MATLAB or python control packages to automate these processes.

There are conditions to this process, one such one being the linear system be controllable, a key failure of gain scheduling.

## 1.3 Iterative Eigenvalue Manipulation

Now that we can know the form of the matrix to solve for and have a method to solve for the gain matrix  $K$ , we must build our pipeline. `#+BEGIN_src`  
text function `getNewControl(f,x)` Given dynamics  $f$ , current state  $x$ , linearize  $f$  about  $x$ , solve for desired eigenvalues, return controller  $u=-K^T x$  `#+END_src`  
text

## 2 General Tools

### 2.1 MATLAB

`syms` MATLAB help page `ode45` MATLAB help page Pole Placement MATLAB Tutorial (Eigenvalue Manipulation)