

# Contents

<b>1</b>	<b>Methodology</b>	<b>1</b>
1.1	Control design . . . . .	1
1.1.1	Dynamics overview . . . . .	1
1.1.2	Control objectives . . . . .	1
1.1.3	Inverse dynamics . . . . .	2
1.1.4	Model Uncertainties . . . . .	3
1.1.5	Sliding Mode . . . . .	4
1.1.6	Sliding Surface Design . . . . .	5
1.1.7	Control Law Design . . . . .	7
1.1.8	Stability analysis . . . . .	9
1.1.9	Summary . . . . .	9

# Chapter 1

## Methodology

### I Control design

#### A. Dynamics overview

Recall the final system of equations for the mathematical model:

$$\begin{cases} M\dot{\bar{v}}^B + C(\bar{v}^B)\bar{v}^B + D(\bar{v}^B)\bar{v}^B + g(\bar{r}^N) = T\phi(u) \\ \dot{\bar{r}}^N = J(\bar{r}^N)\bar{v}^B \end{cases}$$

where  $M = M_B + M_A$ ,  $C(\bar{v}^B) = C_B(\bar{v}^B) + C_A(\bar{v}^B)$

#### B. Control objectives

A main idea of the robust control is in designing such control input  $u$  which ensure that tracking error converges even for the uncertain system.

Underwater robots require precise control systems to navigate and operate effectively in challenging marine environments. These control objectives are crucial for ensuring the robot's stability, accuracy, and responsiveness:

- Position and Orientation Tracking: The robot must accurately follow a de-

sired trajectory, maintaining its position and orientation as intended.

- **Disturbance Rejection:** The robot should be able to withstand external disturbances, such as ocean currents, waves, and sensor noise, to maintain stable tracking performance.
- **Robustness:** The control system should be robust to uncertainties in the robot's dynamics and environmental conditions, ensuring reliable operation even in unpredictable situations.
- **Real-Time Implementation:** The control algorithm should be computationally efficient and able to run in real-time on the robot's embedded system, enabling prompt and effective responses to changing conditions.

A goal is to track the desired trajectory given by  $\bar{\mathbf{r}}_{\text{des}}^B$  thus an error is given by:

$$\tilde{\mathbf{r}}^B = \bar{\mathbf{r}}_{\text{des}}^B - \bar{\mathbf{r}}^B$$

The goal is to find control law such that  $\tilde{\mathbf{r}}^B \rightarrow 0$  as  $t \rightarrow \infty$

### C. *Inverse dynamics*

Inverse dynamics is a nonlinear control technique that provides a trajectory tracking by calculating the required joint actuator torques that need to be generated by the motors to achieve a given trajectory. It relies on exact cancelation of nonlinearities in the robot equation of motion.

The inverse dynamics control approach is directly related to the solution of the inverse dynamics problem. By appropriately inverting the dynamic model, a control law can be constructed which cancels the nonlinear part of the dynamics, decouples the interactions between the regulated variables, and specifies the time characteristics of the decay of the task errors.

Recalling the system dynamics(), we can apply the following outer loop controller to linearize our system:

$$\bar{f}^B = Ma + C(\bar{v}^B)\bar{v}^B + D(\bar{v}^B)\bar{v}^B + g(\bar{r}^N)$$

where  $a$  is outer-loop control to be designed further.

Substitution to the dynamics yields the double integrator system:

$$\begin{aligned} M\dot{\bar{v}}^B + C(\bar{v}^B)\bar{v}^B + D(\bar{v}^B)\bar{v}^B + g(\bar{r}^N) &= Ma + C(\bar{v}^B)\bar{v}^B + D(\bar{v}^B)\bar{v}^B + g(\bar{r}^N) \\ \dot{\bar{v}}^B &= M^{-1}Ma \\ \ddot{\bar{v}}^B &= a \end{aligned}$$

while  $a$  is described in a usual way as  $\ddot{\bar{v}}_{\text{des}}^B - K_p \tilde{r}^B - K_d \tilde{\dot{v}}^B$ .

#### D. Model Uncertainties

**State which parameters are unknown to us - ?**

However now we assume that dynamics is not completely known, as well as there is some disturbance term  $\delta$

Model imprecision may come from actual uncertainty about the plant (e.g., unknown plant parameters), or from the purposeful choice of a simplified representation of the system's dynamics (e.g., modeling friction as linear, or neglecting structural modes in a reasonably rigid mechanical system). From a control point of view, modeling inaccuracies can be classified into two major kinds:

\* structured (or parametric) uncertainties \* unstructured uncertainties (or unmodeled dynamics)

The first kind corresponds to inaccuracies on the terms actually included in the model, while the second kind corresponds to inaccuracies on (i.e., underestimation of) the system order.

As discussed earlier, modeling inaccuracies can have strong adverse effects

on nonlinear control systems. Therefore, any practical design must address them explicitly. Two major and complementary approaches to dealing with model uncertainty are robust and adaptive control. Today we are going to discuss the simple approach to robust control, so-called sliding control methodology.

The dynamics parameter estimates may be imprecise due to unmodeled dynamics and external factors. This means the estimated values might not perfectly match the actual system behavior.

#### *E. Sliding Mode*

Introduce the concept of sliding mode control (SMC) and its advantages for controlling underwater robots.

As discussed in the previous chapter, there are several controller designs available. However, the sliding mode approach suggested by (Spong - ?) is highly regarded as the most sophisticated and frequently implemented one.

Sliding mode control (SMC) is a nonlinear control method that guarantees robust control of systems with uncertainties and disturbances. This technique involves developing a sliding surface within the state space and directing the system's trajectory to slide along this surface (Fig. 1).

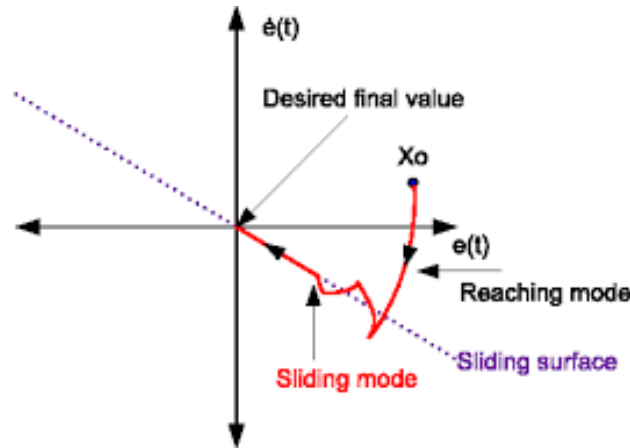


Fig. 1. The general sliding mode scheme

The sliding surface provides robustness to uncertainties and disturbances by ensuring that the system's behavior is insensitive to these factors. This is because the control law is designed to counteract any disturbances or uncertainties that would push the trajectory off the surface. As long as the system's trajectory remains on the sliding surface, the control system will maintain stability and performance, making it an excellent choice for underwater robotics.

Compared to other nonlinear control methods, SMC is a relatively straightforward solution to implement. With a basic understanding of system dynamics and sliding surface design.

#### *F. Sliding Surface Design*

**Explain the concept of a sliding surface and its role in SMC.**

In sliding mode control (SMC), a sliding surface is a hyperplane in the state space that defines the desired system behavior.

The control objective is to force the system's trajectory to slide long this surface. Once the system's trajectory reaches the sliding surface, it will remain on the surface as long as the control law is applied.

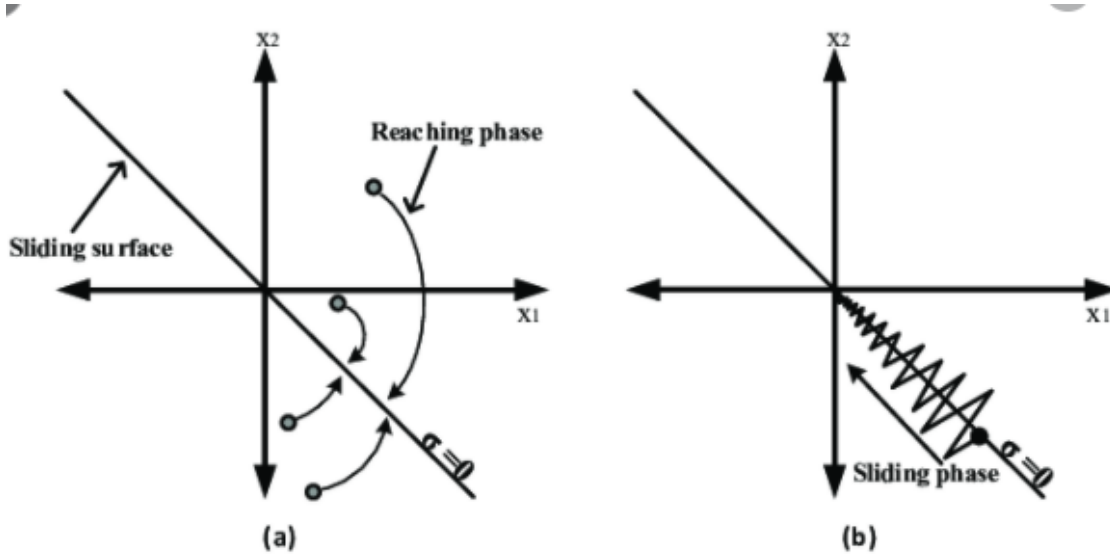


Fig. 2. The phases of sliding mode

Let us define time-varying surface  $\mathcal{S}$  in the the state space  $\mathbb{R}^n$  given by scalar equation  $s(\mathbf{r}^B, t)$ :

$$s(\mathbf{r}^B, t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{\mathbf{r}}^B$$

where  $n$  is the order of the system and  $\lambda$  is a positive scalar?

Analyze the properties of the sliding surface, such as reachability and invariance

The design of the sliding surface is critical for the performance of the SMC system. The sliding surface should be:

- Reachable: The system's trajectory should be able to reach the sliding surface in a finite amount of time.
- Invariant: Once the system's trajectory reaches the sliding surface, it should remain on the surface for all future time.
- Attractive: The control law should attract the system's trajectory to the sliding surface and keep it there.

In order to satisfy the conditions above, the sliding surface is designed to be an invariant set. Invariant sets are sets of states in the state space that, once entered, cannot be exited under the action of the control law.

Let us define  $V = s^2$  as the squared distance to the surface. In order to ensure convergence along all system trajectories one may formulate the following sliding condition:

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|$$

where  $\eta > 0$  define the rate of convergence to the sliding surface.

Satisfying condition or sliding condition, makes the surface an invariant set and implies convergence to  $\tilde{y}$ , since:

$$\left( \frac{d}{dt} + \lambda \right)^{n-1} \tilde{r}^B = 0$$

Applying such transformation yields a new representation of the tracking performance:

$$s \rightarrow 0 \Rightarrow \tilde{r}^B \rightarrow 0 \Rightarrow \tilde{y} \rightarrow 0$$

Meaning, that the problem of tracking  $r^B$  is equivalent to remaining on the sliding surface. Thus the problem of tracking the  $n$ -dimensional vector  $r^B$  can in effect be replaced by a first order stabilization problem in  $s$ .

### *G. Control Law Design*

**Derive the SMC control law for the underwater robot system.**



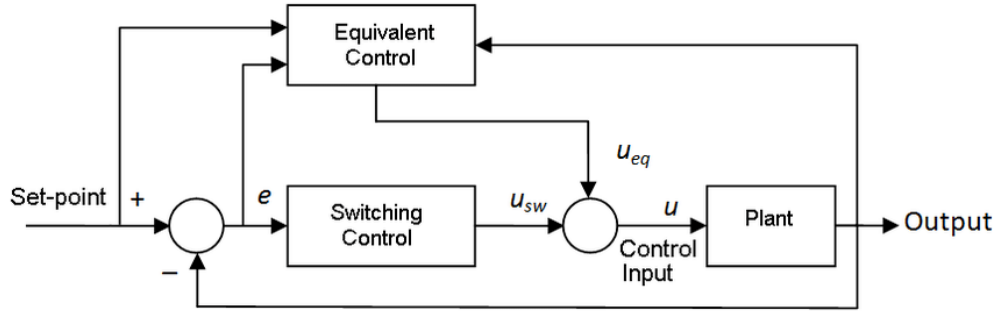


Fig. 3. The sliding mode control scheme

In order to find a minimum of  $s$ , which corresponds to the minimal tracking error, let us take a time derivative:

$$\dot{s} = \dot{\hat{v}}^B + \lambda \dot{\hat{r}}^B = \dot{\hat{v}}^B - \dot{v}^B + \lambda v^B = 0$$

Substituting the system dynamics, it would give us the following equation:

$$\dot{s} = -M^{-1}(C(v^B)v^B + D(v^B)v^B + g(r^N) - f^B) - \dot{v}^B + \lambda v^B = 0$$

Note that inertia matrix  $M$  is always invertible by the construction.

The final expression for control force is:

$$f^B = -Ma + C(v^B)v^B + D(v^B)v^B + g(r^N)$$

where  $a = \dot{v}^B + \lambda v^B$  is outer-loop control. If we substitute () into (), we will get double integrator system in a form:

$$\dot{\hat{v}}^B = a + \eta(\bar{r}^N, \dot{\hat{v}}^B, a)$$

where the uncertainty function  $\eta$  is defined as

$$\eta(\bar{r}^N, \dot{\hat{v}}^B, a) = M^{-1}(\tilde{M}a + \tilde{C}(\bar{v}^B)\bar{v}^B + \tilde{D}(\bar{v}^B)\bar{v}^B + \tilde{g}(\bar{r}^N))$$

In order to ensure global stability of the system, the outer loop control  $a$  designed in a way:

$$a = a_{des}(t) - K_0 v - K_1 r - \delta \alpha$$

where  $\delta\alpha = \begin{cases} \rho \frac{e}{|e|}, & \text{if } \|e\| > 0 \\ 0, & \text{if } \|e\| = 0 \end{cases}$

Define the error  $e$  - ? Does  $e == s$  -?

In order to reduce chattering, the controller above is effectively smoothed using the boundary layer:

$$\delta\alpha = \begin{cases} \rho \frac{e}{\|e\|}, & \text{if } \|e\| \geq \epsilon \\ \frac{\rho}{\epsilon} e, & \text{if } \|e\| < \epsilon \end{cases}$$

where  $\epsilon$  is the boundary thickness.

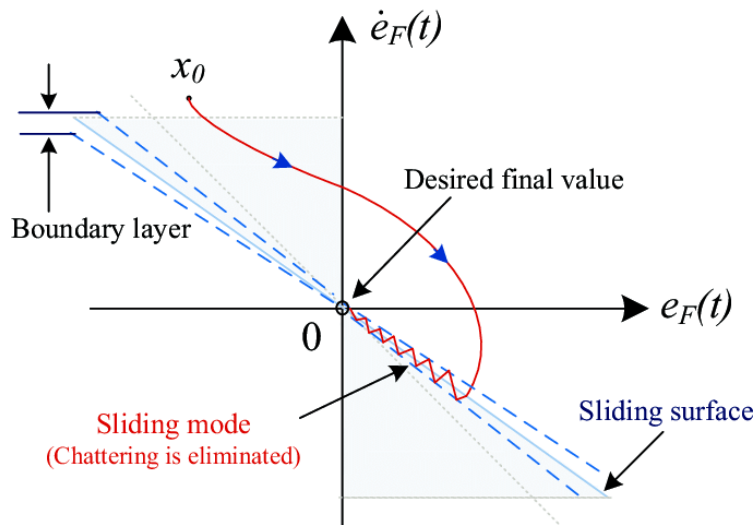


Fig. 4. The sliding mode scheme with boundary layer

*H. Stability analysis*

*I. Summary*