

# Modular Modeling and Trajectory Tracking LQ Regulator for Underwater Robots

Yongyu Chen \* Stefan Sosnowski \* Sandra Hirche \*

\* Chair of Information-oriented Control (ITR), Department of Electrical and Computer Engineering, Technical University of Munich (TUM), Germany  
(e-mail: {yongyu.chen, sosnowski, hirche}@tum.de).

---

**Abstract:** In preparation for formulating the geometric design procedure of an autonomous underwater vehicle (AUV) as an optimization problem, in this paper we introduce a new modular modeling approach. The focus is on the identification of the geometric parametrization and the parameter coupling both for the overall kinematic and kinetic design of the robot as well as the control allocation. In addition, we propose a method to formulate the robot dynamics as a linear multi-input multi-output (MIMO) switched system based on the task-determined trim trajectories and synthesize subsequently a switched linear-quadratic regulator (LQR) controller, which can be checked for global stability. This enables a novel fully automatic task-dependent co-design of an AUV and its controller. Simulation results demonstrate that, in the unconstrained actuator case, the proposed model and control strategy can be successfully applied to perfectly track a set of predefined trim trajectories.

**Keywords:** autonomous vehicles, path planning, marine system, nonlinear control, global stability, linearization, mathematical models

---

## 1. INTRODUCTION

Typical design procedures for AUVs involve standard models for submarine dynamics and an expert adapting these to the specific mission requirements. Often this is an iterative process, as many of the parameters are tightly coupled and immediately influence the dynamics and performance of the vehicle. Furthermore, due to the nonlinear, coupled nature of the vehicle dynamics, changes in the vehicle design parameters, such as a larger hull volume to accommodate more batteries, strongly affect the control design. We therefore aim to solve this challenge as a co-design process of a multi-criterion optimization of the vehicle kinematics and dynamics and a synthesis of a stable controller. The design requirements are specified through a given task, for example following a trajectory within a certain mission duration, which determines the necessary dynamics of the system and the power requirements. We opt for a task representation with trim trajectories as *motion primitives* which can be combined to form arbitrarily complex paths.

An AUV is assembled from a collection of components (hull enclosure, batteries, fins, thrusters, CPU, inertial measurement unit, sensors, etc.). In order to formulate the geometric design procedure of an AUV as an optimization algorithm, a modular model is required. It means that the shape and properties of each component of the underwater robot are parametrized by geometric variables, which in our case will be treated as optimization (i.e., decision) variables. The design requirements (e.g. being buoyancy neutral, low cost and maximum surge velocity) will be integrated in the objective function according to their relationship with these geometric decision variables. To design the underwater robot computationally is equivalent to solving the optimization problems and obtaining the feasible optimal solution. Modular geometry parametrized AUV models have been proposed before, see Nahon (1996) for *A Simplified Dynamics Model for Autonomous Underwater Ve-*

*hicles*, or Chen (2007) and Shuzhe et al. (2011) for *Modular Modeling of Autonomous Underwater Vehicle*. The difference in our approach is that we parametrize the model with an emphasis on the coupling among all geometric decision variables and exposing them directly in the dynamic model formulation, enabling the optimization of the latter.

A computational way to design non-standard multicopters with optimization algorithms for a group of design goals was proposed by Du et al. (2016). Their algorithm iteratively optimizes the geometric (e.g., size, position and orientation) and the controller parameters of the constituent components (propellers, motors and carbon fiber-rods) under different metrics for specific applications. This novel approach enables the user to explore the geometric configuration space and discover robot structure significantly differing from the standard one. This robot geometry would be difficult to be modeled even by a experienced designer. Underwater robots are similar to multicopters in terms of the robot dynamics, both of which are rigid body with 6 degrees of freedom (DoF) influenced by fluid dynamic effects. However, negligible aerodynamic effects become non-negligible in their hydrodynamic equivalents, making the models for the optimization more complex and requiring sophisticated control strategies. Therefore, we adopt a comparable approach to construct a customizable AUV.

To assess the current robot geometry, we use criteria for linear systems such as the condition number of the controllability matrix to judge different configurations of position and orientation of individual actuators. This requires a linearization of the naturally nonlinear and strongly coupled underwater robot dynamics and finding an appropriate equilibrium point is of significance. The concept of trim trajectories is suitable in this case as it corresponds to stable, steady-state, motions of the robot. More precisely, the roll and pitch angles, the dynamics states (linear and angular velocities) and the hydrodynamics

terms depending on the velocities are constant. Trim trajectories used for path planning of aerial robots are described by Sebbane (2014) and Bestaoui et al. (2007). Adopting the Frenet-Serret frame as the desired body frame for the motion, we can further derive the desired Euler angles and the Jacobian matrix. The equilibrium velocities values along trim paths can be computed from the position, Euler angles and the Jacobian matrix so that we can obtain the explicit values of the equilibrium points for the trim trajectory. In conclusion, the equilibrium points contain the full information of the trim trajectory specification.

Using trim trajectories as motion primitives, a complex trajectory could be constructed from a set of trim trajectories based on the mission requirements. For our trajectory-tracking approach, we follow the methodology reported in Silvestre et al. (2002), Kaminer et al. (1998) and Cunha et al. (2005) by defining the error states between the real robot states and the desired ones specified by the trim trajectories. The nonlinear error system can be linearized along the equilibrium point for a specific trim path, which results in a MIMO linear time-invariant (LTI) system. In this way, the trim trajectory specifications are brought into the linearized error dynamics. Thus, we can utilize an linear-quadratic regulator (LQR) to drive the generalized tracking error to zero. Suppose the combined trajectory consists of a set of  $m$  trim trajectories, it is reasonable to formulate the nonlinear robot dynamics as a linear MIMO switched system with  $m$  components and synthesize subsequently a switched LQR controller to realize the path following. A constructive method to ensure the stability of the whole system under arbitrary switching is to find a common quadratic lyapunov function (CQLF) (Shorten et al. (2007)). The state and input weighting matrices can not be arbitrary selected. Their values must be chosen so that a CQLF exists. If we utilize the controllability criteria of these  $m$  linearized systems in the optimization algorithms, the robot geometry would also be determined by the predefined trajectory parameters.

This paper is organized as follows: Section II introduces the modular modeling approach of underwater robot dynamics and identifies the geometric decision variables. In Section III we elaborate how to find the equilibrium points for nonlinear underwater robot dynamics using trim trajectories. The formulation of the trajectory-based error space is discussed and the method for checking the existence of a CQLF is presented. Simulation results illustrating the performance of the proposed control strategy are shown in Section V and Section VI presents the conclusion and future work.

## 2. MODULAR MODELING

The main components determining the shape and the dynamics of an AUVs are the hull enclosing electronic internals and the actuators. The actuators can be further divided into thrusters and control fins. For the sake of simplicity and the computability of the overall model in the optimization process, we make the following assumptions: (i) the hydrodynamic effects on the overall system are dominated by the effects on the hull and fins, those of the thrusters are negligible; (ii) the moment of inertia of the system is assumed to be merely determined by the hull components, fins and thrusters are neglected.

### 2.1 Modeling of Hull Components

A world inertial frame  $\{i\} = (x_i \ y_i \ z_i)$  describes the environment of the underwater robot which is fixed in space. The body-fixed

reference frame  $\{b\} = (x_b \ y_b \ z_b)$  for the whole robot is an accelerated coordinate frame fixed to the robot. The origin  $o_b$  is usually chosen to coincide with the geometric center of the hull which will be referred to as  $CO$ . The longitudinal axis  $x_b$  is directed from aft to fore, the transversal axis  $y_b$  is directed to starboard and the normal axis  $z_b$  orientates from top to bottom, see Fig. 1.

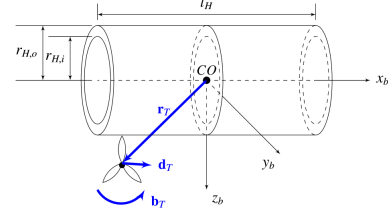


Fig. 1. Hull and thruster geometry

The main body typically embeds the navigation-, the sensory- and the power supply-system. The shell is modeled as a hollow cylinder. We use three parameters to describe the dimensions: the inner radius  $r_{H,i}$ , the outer radius  $r_{H,o}$  and the hull length  $l_H$ . Its mass  $m_H$  is then calculated as  $m_H = \rho_H \pi (r_{H,o}^2 - r_{H,i}^2) l_H$ , with  $\rho_H$  being the density of the hull material, and the moment of inertia with respect to its center of gravity can be calculated as follows:

$$I_H = \text{diag}([I_{xx,H}, I_{yy,H}, I_{zz,H}])$$

where  $I_{xx,H} = 1/2 m_H (r_{H,o}^2 + r_{H,i}^2)$  and  $I_{yy,H} = I_{zz,H} = 1/4 m_H (r_{H,o}^2 + r_{H,i}^2 + l_H^2/3)$ , due to its rotational symmetry along the  $x_b$ -axis and the symmetry with respect to the  $z_b y_b$ -plane. In the following, we use  $r_H$  to represent  $r_{H,o}$ . Within the hull, we model the battery as a solid cylinder with length  $l_{bat}$  and radius  $r_{bat}$ , the mass  $m_{bat} = \rho_{bat} \pi r_{bat}^2 l_{bat}$  and the moment of inertia  $I_{bat}$  with respect to the battery center of gravity is:

$$I_{bat} = \text{diag}([I_{xx,bat}, I_{yy,bat}, I_{zz,bat}])$$

where  $I_{xx,bat} = I_{yy,bat} = 1/12 m_{bat} (3r_{bat}^2 + l_{bat}^2)$  and  $I_{zz,bat} = 1/2 m_{bat} r_{bat}^2$ . For all other electronic devices, we assume they have a constant mass  $m_d$ , volume  $V_d$  and moment of inertia  $I_d$  with respect to the robot body axes  $\{b\}$ . For different underwater robot prototypes, the main customizable decision variables for the hull are its length  $l_H$  and radius  $r_H$  to enclose all internal devices and influence buoyancy. Therefore, we choose them as the decision variables of hull geometry and obtain the following definition:

**Definition 1.** The geometric decision parameters for the hull  $\mathfrak{d}_H$  are defined as  $\mathfrak{d}_H = \{l_H, r_H \mid l_H \in \mathbb{R}^+, r_H \in \mathbb{R}^+\}$ , where  $l_H$  denotes the length of the hull cylinder and  $r_H$  is the radius of the hull cylinder.

### 2.2 Modeling of Thrusters

As shown in Fig. 1, each thruster is parameterized by the motor orientation  $\mathbf{d}_T$  (unit vector), the motor position  $\mathbf{r}_T$  in the body frame  $\{b\}$  and the motor spin direction  $b_T \in \{-1, 1\}$ , where -1 means counterclockwise and 1 represents clockwise. We use  $u_T$  and  $m_{T,r}$  to denote the magnitude of the thrust force and torque generated by the propeller rotation, respectively. A simple one-state model for the propeller is proposed by (Yoerger et al. (1990)). If the propeller rotates in a steady state, the torque  $m_{T,r}$  due to the rotation of a propeller is proportional to the thrust force. Thus, we can represent their relationship as  $m_{T,r} = \lambda_T u_T$ , where  $\lambda_T$  is the torque-force ratio. Taking the thruster motor

orientation and the spin direction into consideration, we can obtain the rotation torque vector in  $\{b\}$  as  $\mathbf{m}_{T,r} = b_T \lambda_T u_T \mathbf{d}_T$ . In addition, the thrust force will produce the thrust torque  $\mathbf{m}_{T,t}$  relative to the center of gravity  $CG$  of the complete robot.  $CG$  is assumed to be located at the origin  $CO$  of the body frame  $\{b\}$ . Then, the thrust torque can be calculated as  $\mathbf{m}_{T,t} = \mathbf{r}_T \times u_T \mathbf{d}_T$ , where  $\mathbf{r}_T = (x_T \ y_T \ z_T)^T \in \mathbb{R}^3$  is the position vector of the thruster in the body frame  $\{b\}$ . Combining the thrust torque and the rotation torque, we express the total torque produced by a thruster as  $\mathbf{m}_T = \mathbf{m}_{T,r} + \mathbf{m}_{T,t}$ . The force direction is determined by the thruster motor orientation  $\mathbf{d}_T$  and the force vector  $\mathbf{f}_T$  is written as  $\mathbf{f}_T = u_T \mathbf{d}_T \in \mathbb{R}^6$ . The generalized force vector  $\boldsymbol{\tau}_T$  generated by a thruster is defined as  $\boldsymbol{\tau}_T = (\mathbf{f}_T \ \mathbf{m}_T)^T \in \mathbb{R}^6$ . If we treat the thrust force  $u_T$  as the control input, we can define a unique vector  $B_T$  mapping the thrust  $u_T$  to the generalized force  $\boldsymbol{\tau}_T$  as

$$B_T = \begin{bmatrix} \mathbf{d}_T \\ b_T \lambda_T \mathbf{d}_T + \mathbf{r}_T \times \mathbf{d}_T \end{bmatrix} \in \mathbb{R}^6. \quad (1)$$

The generalized force produced by thrust force can thus be calculated as  $\boldsymbol{\tau} = B_T u_T$ . For the thrusters, we choose the geometric decision variables as follows:

**Definition 2.** The geometric parameter set for the thrusters  $\mathfrak{d}_P$  is defined as  $\mathfrak{d}_P = \{b_T, \mathbf{d}_T, \mathbf{r}_T \mid b_T \in \{-1, 1\}, \mathbf{d}_T \in \mathbb{R}^3, \mathbf{r}_T \in \mathbb{R}^3\}$ , where  $b_T$  denotes the motor spin direction,  $\mathbf{d}_T$  is the motor orientation and  $\mathbf{r}_T$  represents the motor position.

### 2.3 Modeling of Control Fins

The fins are approximated as rectangle with length  $a_F$ , width  $b_F$  and negligible thickness. We assume the side edge  $b_F$  is directly attached to the hull surface. The fins have a controllable DoF around an axis parallel to  $a_F$  in the midsection of  $b_F$ . We represent the position and the orientation of the fins in the hull cylindrical coordinate system as illustrated in Fig. 2, where  $x_F$  is defined as the  $x_b$ -coordinate of the fin geometric center in robot body frame  $\{b\}$ . Then we rotate the  $x_b z_b$ -plane in the counterclockwise direction around  $x_b$  by angle  $\gamma_F$  until the fin geometric center is located in this plane. As a result, the fin

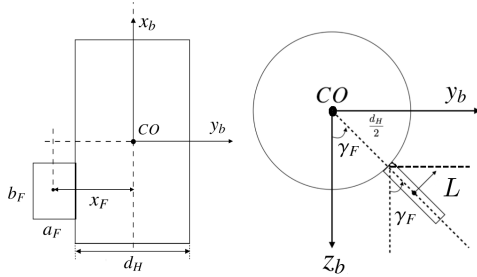


Fig. 2. Definition of fin geometry

geometric center  $\mathbf{r}_F$  in the robot body frame  $\{b\}$  can be written as

$$\mathbf{r}_F = \begin{bmatrix} x_F \\ 0.5(d_H + a_F) \sin(\gamma_F) \\ 0.5(d_H + a_F) \cos(\gamma_F) \end{bmatrix} \in \mathbb{R}^3. \quad (2)$$

The unit vectors  $\mathbf{e}_1 = (0 \ 0 \ 1)^T$ ,  $\mathbf{e}_2 = (0 \ 1 \ 0)^T$  and  $\mathbf{e}_3 = (1 \ 0 \ 0)^T$  are the standard orthonormal basis of the robot body frame. We assume the fin velocity  $\mathbf{v}_{fin}$  is equal to the robot velocity  $\mathbf{v}$ . For each fin, the lift  $\mathbf{L}$  and drag  $\mathbf{D}$  of fins can be calculated as

$$\mathbf{L} = 0.5 \rho a_F b_F C_L(\alpha) \|\mathbf{v}\|^2 \begin{bmatrix} 0 \\ \cos(\gamma_F) \\ -\sin(\gamma_F) \end{bmatrix} \in \mathbb{R}^3, \quad (3)$$

and

$$\mathbf{D} = 0.5 \rho a_F b_F C_D(\alpha) \|\mathbf{v}\|^2 \mathbf{e}_1 \in \mathbb{R}^3, \quad (4)$$

where  $a_F$  and  $b_F$  are the length and the width of the fin, respectively.  $C_L$  is the lift coefficient and  $C_D$  represents the drag coefficient, both of which are dependent on the angle of attack  $\alpha$ . Assuming the robot mainly moves forward, the surge velocity  $u$  is dominant with respect to the sway  $v$  and heave velocity  $w$ . Consequently, we can approximate  $\|\mathbf{v}\|$  with  $u$ . Suppose the robot has  $n_f$  fins, then the resultant force vector produced by all fins can be calculated as

$$\mathbf{f}_F(\alpha_i) = \begin{bmatrix} \sum_{i=1}^{n_f} L_i(\alpha_i)^T \mathbf{e}_1 + \sum_{i=1}^{n_f} D_i(\alpha_i)^T \mathbf{e}_1 \\ \sum_{i=1}^{n_f} L_i(\alpha_i)^T \mathbf{e}_2 + \sum_{i=1}^{n_f} D_i(\alpha_i)^T \mathbf{e}_2 \\ \sum_{i=1}^{n_f} L_i(\alpha_i)^T \mathbf{e}_3 + \sum_{i=1}^{n_f} D_i(\alpha_i)^T \mathbf{e}_3 \end{bmatrix} \in \mathbb{R}^3. \quad (5)$$

The resultant torque vector is calculated as

$$\boldsymbol{\tau}_F(\alpha_i) = \sum_{i=1}^{n_f} \mathbf{r}_{F,i} \times L_i(\alpha_i) + \sum_{i=1}^{n_f} \mathbf{r}_{F,i} \times D_i(\alpha_i) \quad (6)$$

We regard the lift force as the input force and the drag as a disturbance. For small angles of attack  $\alpha$  (about  $15^\circ - 20^\circ$ ), it is reasonable to approximate the lift and drag coefficient as  $C_L(\alpha) = c_L \alpha$  and  $C_D(\alpha) = c_D \alpha^2$ , where  $c_L$  and  $c_D$  are fin-specific parameters measured from experiments, respectively. Furthermore, the angle of attack  $\alpha$  can be approximated by the mechanical angle of rotation  $\delta_F$  in the DoF of the fins. We treat  $\delta_F$  as the control input for the fin actuators and define a mapping vector for each fin:

$$B_F = \begin{bmatrix} 0 \\ 0.5 \rho c_L a_F b_F u^2 \cos(\gamma_F) \\ 0.5 \rho c_L a_F b_F u^2 \sin(\gamma_F) \\ -0.25 \rho c_L a_F b_F u^2 \sin(d_H + a_F) \\ 0.5 \rho c_L a_F b_F u^2 x_F \sin(\gamma_F) \\ 0.5 \rho c_L a_F b_F u^2 x_F \cos(\gamma_F) \end{bmatrix} \in \mathbb{R}^6 \quad (7)$$

The fin mapping vector  $B_F$  depends not only on the geometric parameters but also on the state of the rigid body due to the existence of the surge velocity term, while the thruster mapping vector is only geometry-determined. The generalized force generated by the fins can be written as:  $\boldsymbol{\tau}_F = (\mathbf{f}_F \ \mathbf{m}_F)^T = B_F \delta_F \in \mathbb{R}^6$ . We select  $x_F$  and  $\gamma_F$  as decision variables for the fins:

**Definition 3.** The geometric parameter set for the fins  $\mathfrak{d}_F$  is defined as  $\mathfrak{d}_F = \{x_F, \gamma_F \mid x_F \in \mathbb{R}, \gamma_F \in (-\pi, \pi)\}$ , where  $x_F$  denotes the  $x_b$ -coordinate of the fin geometric center in the body frame  $\{b\}$ ,  $\gamma_F$  is the fin placement angle.

### 2.4 Estimation of the Hydrodynamic Effects

Hydrodynamic effects (added mass and hydrodynamic damping) are the distinct feature for underwater vehicles. Strictly speaking, each component (hull, thrusters and fins) contributes to the hydrodynamic forces and moments. They depend not only on the robot geometry but also on the inflow fluid velocity. Normally, the underwater robot dynamics are defined in the robot body frame  $\{b\}$ . However, the hydrodynamic effect of each module is calculated with respect to its own body frame. The calculated hydrodynamic effects need be transformed into the robot body frame  $\{b\}$  according to the location of each module in  $\{b\}$  to build the robot dynamics. This

transformation results in nonlinearities and couplings between robot states and the actuator geometric variables. We assume that the added masses of thrusters and fins are negligible and only the added mass of the cylindrical hull is taken into consideration. The cylindrical hull has three planes of symmetry. The structure of the added mass inertia matrix  $M_A$  and the added mass Coriolis matrix  $C_A$  for a vehicle completely submerged in the water can be found in Vollmayr et al. (2014). The added mass coefficients can be calculated from the hull geometric parameters:  $X_{\dot{u}} = -0.1m_H$ ,  $Y_{\dot{v}} = Z_{\dot{w}} = -\pi\rho r_H^2 l_H$ ,  $K_{\dot{p}} = 0$ ,  $M_{\dot{q}} = N_{\dot{r}} = -1/12\pi\rho r_H^2 l_H^3$ . The damping matrix  $D = \text{diag}([-X_{u|u}|u|, -Y_{v|v}|v|, -Z_{w|w}|w|, -K_{p|p}|p|, -M_{q|q}|q|, -N_{r|r}|r|])$ , the damping coefficients can be also calculated by the hull geometric parameters (see Antonelli (2003)). The damping coefficient in the  $x_b$ -direction due to the viscous force acting on the frontal area is given by  $X_{u|u} = -1/2\rho\pi C_D r_H^2$ . The damping coefficients in the  $y_b$ - and  $z_b$ -axes due to the viscous effects on the side projected area of the hull are given as  $Y_{v|v} = Z_{w|w} = -1/2\rho C_D 2r_H l_H$ . The hull has no contribution to the roll moment, hence  $K_{p|p} = 0$ . The pitch and yaw damping coefficients are equal due to the hull symmetry and given as  $M_{q|q} = N_{r|r} = -1/12\rho C_D r_H l_H^3$  (see Tang (1999)). In this way, we establish the relationship between the explicit values of hydrodynamic terms and the geometric decision variables.

### 2.5 Dynamic Model

The state variables of underwater robots can be divided into two groups, i.e., kinematic and dynamic states. Dynamic states consist of three linear and three angular velocities represented in the robot body frame  $\{b\}$ :  $\mathbf{x}_{dyn} = (\mathbf{v} \ \boldsymbol{\omega})^T = (u \ v \ w \ p \ q \ r)^T \in \mathbb{R}^6$ . Kinematic states consist of the position and orientation of the underwater robot in the inertial world frame  $\{i\}$ :  $\mathbf{x}_{kin} = (\mathbf{p} \ \boldsymbol{\eta})^T = (x \ y \ z \ \phi \ \theta \ \psi)^T \in \mathbb{R}^6$ . The complete states are defined as  $\mathbf{x} = (\mathbf{x}_{dyn} \ \mathbf{x}_{kin})^T \in \mathbb{R}^{12}$ . The standard underwater robot dynamic equation is given by

$$M \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + C(\mathbf{v}, \mathbf{w}) \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} + D(\mathbf{v}, \mathbf{w}) \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}. \quad (8)$$

Based on the modular modeling, each term in (8) can be uniquely computed from the predefined groups of geometric decision variables.

$$M := M_{RB}(\mathfrak{d}_H, \mathfrak{d}_P, \mathfrak{d}_F) + M_A(\mathbf{x}_{dyn}, \mathfrak{d}_H) \quad (9)$$

is the generalized mass matrix of the underwater robot with  $M_{RB}$  and  $M_A$  representing the rigid body mass matrix and the added mass matrix, respectively. Furthermore,

$$C(\mathbf{v}, \mathbf{w}) := C_{RB}(\mathbf{x}_{dyn}, \mathfrak{d}_H, \mathfrak{d}_P, \mathfrak{d}_F) + C_A(\mathbf{x}_{dyn}, \mathfrak{d}_H) \quad (10)$$

is the Coriolis-centripetal matrix (including added mass effects),  $D(\mathbf{x}_{dyn}, \mathfrak{d}_H)$  is the damping matrix and  $\mathbf{g}(\boldsymbol{\eta})$  is the vector of gravitational/buoyancy forces and moments, which can be written in the geometry-dependent form as  $\mathbf{g}(\mathbf{x}_{kin}, \mathfrak{d}_H, \mathfrak{d}_P, \mathfrak{d}_F)$ .  $\boldsymbol{\tau}$  is the generalized force vector generated by thrusters and control surfaces (fins). We use  $\mathbf{u}_T$  and  $\mathbf{u}_F$  to denote the control inputs generated by thrusters and fins, respectively. Suppose the robot has  $n_t$  thrusters and  $n_f$  fins, then the thruster input vector can be written as  $\mathbf{u}_T = (u_{T,1}, \dots, u_{T,n_t})^T \in \mathbb{R}^{n_t \times 1}$  and the fin input vector is defined as  $\mathbf{u}_F = (\delta_{F,1}, \dots, \delta_{F,n_f})^T \in \mathbb{R}^{n_f \times 1}$ . The complete input vector for the robot system is defined as  $\mathbf{u} = (\mathbf{u}_T \ \mathbf{u}_F)^T \in \mathbb{R}^{(n_t+n_f) \times 1}$ . For each actuator, there is a corresponding input mapping vector and we can formulate the thruster input matrix as  $B_{a,T} = (B_{T,1} \dots B_{T,n_t}) \in \mathbb{R}^{6 \times n_t}$ . The fin input matrix is defined as  $B_{a,F} = (B_{F,1} \dots B_{F,n_f}) \in \mathbb{R}^{6 \times n_f}$ .

Stacking them together, we get the input mapping matrix  $B_a = (B_{a,T} \ B_{a,F})^T \in \mathbb{R}^{6 \times (n_t+n_f)}$ . Thus, we obtain the relationship  $\boldsymbol{\tau} = B_a \mathbf{u}$ . Given a robot prototype with known geometric data of all components, we can directly construct the dynamic equation for analysis.

## 3. TRAJECTORY-DEPENDENT ERROR SYSTEM

In this section, we introduce the transformation of the underwater dynamic model into an error system when the robot tracks a trim trajectory. Moreover, the linearization of the error dynamics is discussed.

### 3.1 Frenet-Serret Frame

When the robot moves along a smooth curve  $\mathbf{r}(t) = (x(t) \ y(t) \ z(t))^T$  in the three-dimensional Euclidean space  $\mathbb{R}^3$ , its motion can be characterized by three unit vectors:  $T^i(t)$ ,  $N^i(t)$  and  $B^i(t)$ .  $T^i(t)$  is a unit vector tangent to the curve denoting the motion direction, which can be calculated as the changing rate of the curve vector with respect to its arclength  $s$ :

$$T^i(t) = \frac{d\mathbf{r}}{ds} = \frac{\mathbf{r}'(t)}{\|\mathbf{r}'(t)\|}. \quad (11)$$

The normal vector  $N^i(t)$  is defined as the derivative of  $T^i$  with respect to the arclength  $s$  of the curve, divided by its length:

$$N^i(t) = \frac{\frac{dT^i(t)}{ds}}{\left\| \frac{dT^i(t)}{ds} \right\|} = \frac{(T^i)'(t)}{\|(T^i)'(t)\|} = \frac{\mathbf{r}'(t) \times \mathbf{r}''(t) \times \mathbf{r}'(t)}{\|\mathbf{r}'(t)\| \|\mathbf{r}''(t) \times \mathbf{r}'(t)\|}. \quad (12)$$

The binormal vector  $B^i_T$  is the cross product of the normal vector  $N^i(t)$  and the tangent vector  $T^i(t)$  and is calculated as

$$B^i(t) = T^i(t) \times N^i(t) = \frac{\mathbf{r}''(t) \times \mathbf{r}'(t)}{\|\mathbf{r}''(t) \times \mathbf{r}'(t)\|}. \quad (13)$$

### 3.2 Trim Trajectory

The concept of trim trajectories originates in the path planning for aircrafts. It corresponds to a curve  $\mathbf{p}_{\mathcal{T}} = (x_{\mathcal{T}} \ y_{\mathcal{T}} \ z_{\mathcal{T}})^T \in \mathbb{R}^3$  that the robot can follow in a steady-state equilibrium. A trim trajectory  $\mathbf{p}_{\mathcal{T}}$  is parametrized by three parameters  $\mathbf{p}_{\mathcal{T}} : (\|\mathbf{v}_{\mathcal{T}}\| \ \phi_{\mathcal{T}} \ \gamma_{\mathcal{T}})^T$ , where  $\mathbf{v}_{\mathcal{T}}$  is the trim speed. We choose the aforementioned Frenet-Serret frame as the desired body frame for the robot moving along the trim trajectory curve and denote it as  $\{\mathcal{T}\}$ . The desired orientation on basis of  $\{\mathcal{T}\}$  for the trim trajectory is represented as  $\boldsymbol{\eta}_{\mathcal{T}} = (\phi_{\mathcal{T}} \ \theta_{\mathcal{T}} \ \psi_{\mathcal{T}})^T$ . For  $\{\mathcal{T}\}$ , the trim speed is equal to the desired surge velocity  $u$ . The angle  $\gamma_{\mathcal{T}}$  is the flight path angle. The trim yaw angle is given by  $\phi_{\mathcal{T}}(t) = \dot{\phi}_{\mathcal{T}} t + \phi_0$ , where  $\dot{\phi}_{\mathcal{T}} \in \mathbb{R}$  is the constant yaw rate and  $\phi_0 \in [0, 2\pi)$  is the initial yaw angle. Along the trim trajectory, the roll and pitch angle remain unchanged, i.e.,  $\dot{\phi}_{\mathcal{T}} = 0$  and  $\dot{\theta}_{\mathcal{T}} = 0$ . A trim trajectory segment corresponds to a helix with radius  $\|\mathbf{v}_{\mathcal{T}}\| \dot{\psi}_{\mathcal{T}}^{-1} \cos(\gamma_{\mathcal{T}})$  and is represented as

$$\mathbf{p}_{\mathcal{T}}(t) = \|\mathbf{v}_{\mathcal{T}}\| \dot{\psi}_{\mathcal{T}}^{-1} \cos(\gamma_{\mathcal{T}}) \begin{bmatrix} \sin(\psi_{\mathcal{T}}(t) - \psi_v) - \sin(\psi_0 - \psi_v) \\ \cos(\psi_{\mathcal{T}}(t) - \psi_v) + \cos(\psi_0 - \psi_v) \\ -\dot{\psi}_{\mathcal{T}} \tan(\gamma_{\mathcal{T}}) t \end{bmatrix} + \mathbf{p}_0, \quad (14)$$

where  $\mathbf{p}_0 = (x_0 \ y_0 \ z_0)^T \in \mathbb{R}^3$  is the initial trim position. The transformation matrix from the Frenet-Serret frame  $\{\mathcal{T}\}$  to the inertial world frame  $\{i\}$  is defined as  $\mathcal{R}_{\mathcal{T}} := (T^i(t) \ N^i(t) \ B^i(t))$ .

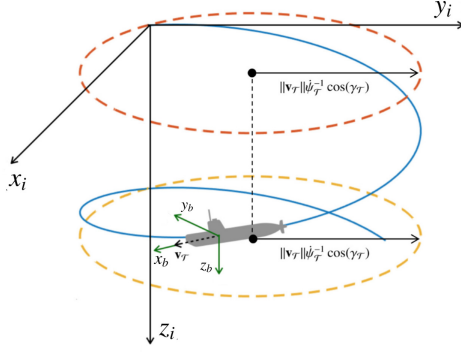


Fig. 3. Trim trajectory

The desired trim roll and pitch angle can be inferred from the Frenet-Serret frame setting as  $\theta_T = -\sin^{-1}(T^i(t))$  and  $\phi_T = -\sin^{-1}(\sec(\theta_T) \cdot N^i(t))$ , respectively. Stacking the two dynamic-affecting angles together we define  $\eta_{p,T} = (\phi_T \ \theta_T)^T$ . With the value of  $\phi_T$  and  $\theta_T$ , we can obtain the angular velocity transformation matrix  $Q_T$ . Thus, the desired velocities along the trim trajectory can be calculated as

$$\begin{bmatrix} v_T \\ w_T \end{bmatrix} = \begin{bmatrix} R_T^{-1} & 0 \\ 0 & Q_T^{-1} \end{bmatrix} \begin{bmatrix} p_T \\ \eta_T \end{bmatrix}. \quad (15)$$

All the variables affecting the underwater robot dynamics ( $v_T$ ,  $w_T$ ,  $\eta_{p,T}$ ), see (8), stay constant. They correspond to the equilibrium point for motion along the trim trajectory. The equilibrium input corresponding to  $\dot{v}_T = 0$  and  $\dot{w}_T = 0$  can be calculated as

$$\tau_{d,T} = C_{RB}(x_{dyn,T}, r_T, d_H, d_F) \begin{bmatrix} v_T \\ w_T \end{bmatrix} + C_A(x_{dyn,T}, d_H) v_T + D(x_{dyn,T}, d_H) v_T + g(x_{kin,T}, d_F, r_T, d_H). \quad (16)$$

The underwater dynamics can be linearized for explicit values of  $v_T$ ,  $w_T$ ,  $\eta_{p,T}$  and  $u_T$ . The state space form of the underwater robotic system is given by

$$\frac{d}{dt} \begin{bmatrix} x_{dyn} \\ x_{kin} \end{bmatrix} = \begin{bmatrix} f_{dyn}(x_{dyn}, x_{kin}) \\ f_{kin}(x_{dyn}, x_{kin}) \end{bmatrix} + \begin{bmatrix} g_{dyn}(x_{dyn}, u) \\ 0 \end{bmatrix}. \quad (17)$$

Equivalently, they can be written in the following form:

$$\frac{d}{dt} v = f_v(v, w) + f_v^\eta(\eta_p) + g_v(v, u) \quad (18)$$

$$\frac{d}{dt} w = f_w(v, w) + f_w^\eta(\eta_p) + g_w(v, u) \quad (19)$$

$$\frac{d}{dt} p = R(\eta) v \quad (20)$$

$$\frac{d}{dt} \eta = Q(\eta_p) \omega. \quad (21)$$

where  $\eta_p = (\phi \ \theta)^T$  denotes the partial kinematic states affecting the underwater robot dynamics appearing in the restoring term  $g(\eta)$ ,  $R$  is the linear velocity transformation matrix from  $\{b\}$  to  $\{i\}$  and  $Q$  is the transformation matrix relating the body fixed angular velocity  $w$  to Euler angle rates. We now need to convert (8) into the state space form, see (18) and (19).

The first three rows of the matrix  $B_a$  are denoted by  $B_{a,f} \in \mathbb{R}^{3 \times (n_t + n_f)}$  and the last three rows of the matrix are represented by  $B_{a,m} \in \mathbb{R}^{3 \times (n_t + n_f)}$ . The generalized force generated by the actuators can be represented as  $\tau = B_a u$ . Separately written,  $f = B_{a,f} u$  and  $m = B_{a,m} u$ , where  $f$  is the force vector along the  $x_b$ -,  $y_b$ - and  $z_b$ -axes of  $\{b\}$  and  $m$  is the moment vector about those axes. Hence, for our modeling approach, we get  $g_v(v, u) = M^{-1} B_{a,f} u$  and  $g_w(v, u) = M^{-1} B_{a,m} u$ . The rigid

body Coriolis-centripetal matrix  $C_{RB}(v, w)$  can be separated into  $C_{RB,v}$  and  $C_{RB,w}$  corresponding to the first and last three rows of the Coriolis matrix  $C_{RB}$ , respectively. Similarly, we assign the first three lines of the added mass Coriolis matrix  $C_A$  to  $C_{A,v}(v, w)$  and the last three lines to  $C_{A,w}(v, w)$ . The restoring vector  $g(\eta)$  can also be separated in the same way into  $g_v(\eta) \in \mathbb{R}^3$  that is the first three elements of  $g(\eta)$  and  $g_w(\eta) \in \mathbb{R}^3$  corresponding to the last three elements. To sum up, we can obtain the following relationship between the terms in the underwater robot dynamics and the state-space terms:

$$f_v(v, w) = -M^{-1}(C_{RB,v} + C_{A,v} + D_v), \quad (22)$$

$$f_w(v, w) = -M^{-1}(C_{RB,w} + C_{A,w} + D_w), \quad (23)$$

$$f_v^\eta(\eta_p) = -M^{-1} g_v(\eta), \quad (24)$$

$$f_w^\eta(\eta_p) = -M^{-1} g_w(\eta). \quad (25)$$

### 3.3 Nonlinear Transformation and Error Dynamics

Fig. 4 shows the relationship between the inertial world frame  $\{i\}$ , the robot body frame  $\{b\}$  and the Frenet-Serret frame along the trim curve  $\{T\}$ . In the ideal case, the robot body frame  $\{b\}$  should coincide with the frame  $\{T\}$  for perfect tracking, but it is not necessarily the case in reality. Therefore, we use  $R_E$  to denote the transformation matrix from  $\{b\}$  to  $\{T\}$  for linear velocities and position and  $Q_E$  for angular velocities. Based on these definitions, we have the following relationship  $R_E = R_T^{-1} R$  and  $Q_E = Q_T^{-1} Q$ . We adopt the following transformation for the state errors:  $v_E = v - v_T$ ,  $w_E = w - w_T$ ,  $p_E = R^{-1}(p - p_T)$  and  $\eta_E = R^{-1}(\eta - \eta_T)$ , which can be interpreted as the errors between the desired trim states and the real robot states as measured by sensors in the robot body frame  $\{b\}$ . We define the error dynamics input as  $u_E = u - u_T$ , where  $u$  is the control input from the actuators and  $u_T$  is the desired trim actuator input, which is calculated as  $u_T = B_a^{-1} \tau_{d,T}$ .

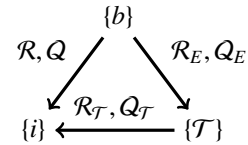


Fig. 4. Frame relationship

### 3.4 Linearization of Error System

In the previous section, we have derived the equilibrium point for the motion of the AUV along a trim trajectory and defined the error system for tracking this trajectory. It is common to control a nonlinear system by linearizing it at an equilibrium point. Moreover, we can use properties of the linearized system as performance criteria for finding an optimal geometry, e.g., the controllability matrix. Thus, in this section, we perform the linearization of (22), (23), (24) and (25) at  $(v_T, w_T, \eta_{p,T}, u_T)$ ,

$$\frac{d}{dt} \delta v_E = A_v^v \delta v_E + A_v^w \delta w_E + A_v^\eta \delta \eta_E + B_v \delta u \quad (26)$$

$$\frac{d}{dt} \delta w_E = A_w^v \delta v_E + A_w^w \delta w_E + A_w^\eta \delta \eta_E + B_w \delta u \quad (27)$$

where

$$\begin{cases} A_v^v = \frac{\partial}{\partial v} [f_v(v, w) + g_v(v, w, u)] \\ A_v^w = \frac{\partial}{\partial w} [f_v(v, w) + g_v(v, w, u)] \end{cases} \quad (28)$$

$$\begin{cases} A_v^v = \frac{\partial}{\partial v} [f_v(v, w) + g_v(v, w, u)] \\ A_v^w = \frac{\partial}{\partial w} [f_v(v, w) + g_v(v, w, u)] \end{cases} \quad (29)$$

$$\begin{cases} A_v^\eta = \frac{\partial}{\partial \eta} f_v^\eta(\eta_p) \\ B_v = \frac{\partial}{\partial u} g_v(v, w, u) \end{cases} \quad (30)$$

$$\begin{cases} A_v^\eta = \frac{\partial}{\partial \eta} f_v^\eta(\eta_p) \\ B_v = \frac{\partial}{\partial u} g_v(v, w, u) \end{cases} \quad (31)$$

and

$$\begin{cases} A_w^v = \frac{\partial}{\partial v} [f_w(v, w) + g_w(v, w, u)] \\ A_w^w = \frac{\partial}{\partial w} [f_w(v, w) + g_w(v, w, u)] \end{cases} \quad (32)$$

$$\begin{cases} A_w^v = \frac{\partial}{\partial v} [f_w(v, w) + g_w(v, w, u)] \\ A_w^w = \frac{\partial}{\partial w} [f_w(v, w) + g_w(v, w, u)] \end{cases} \quad (33)$$

$$\begin{cases} A_w^\eta = \frac{\partial}{\partial \eta} f_w^\eta(\eta_p) \\ B_w = \frac{\partial}{\partial u} g_w(v, w, u) \end{cases} \quad (34)$$

$$\begin{cases} A_w^\eta = \frac{\partial}{\partial \eta} f_w^\eta(\eta_p) \\ B_w = \frac{\partial}{\partial u} g_w(v, w, u) \end{cases} \quad (35)$$

The calculation of linearizing the position and Euler angle error can be found in Silvestre et al. (2002). The resulting linearization is

$$\begin{cases} \frac{d}{dt} \delta p_E = \delta v_E - S(v_T) \delta p_E - S(v_T) \delta \eta_E \\ \frac{d}{dt} \delta \eta_E = \delta w_E - S(w_T) \delta \eta_E \end{cases} \quad (36)$$

$$\begin{cases} \frac{d}{dt} \delta p_E = \delta v_E - S(v_T) \delta p_E - S(v_T) \delta \eta_E \\ \frac{d}{dt} \delta \eta_E = \delta w_E - S(w_T) \delta \eta_E \end{cases} \quad (37)$$

where  $S$  is a skew-symmetrical matrix satisfying  $S = S^T$  and defined as

$$S(\zeta) = -S^T(\zeta) = \begin{bmatrix} 0 & -\zeta_3 & \zeta_2 \\ \zeta_3 & 0 & -\zeta_1 \\ -\zeta_2 & \zeta_1 & 0 \end{bmatrix} \quad (38)$$

for an arbitrary vector  $\zeta = (\zeta_1 \ \zeta_2 \ \zeta_3)^T$ . In total, the linearized error dynamics are

$$\begin{cases} \frac{d}{dt} \delta x_{dyne} = A_d^d \delta x_{dyne} + A_k^d \delta x_{kinE} + B u \\ \frac{d}{dt} \delta x_{kinE} = \delta x_{dyne} + A_k^k \delta x_{kinE} \end{cases} \quad (39)$$

$$\begin{cases} \frac{d}{dt} \delta x_{dyne} = A_d^d \delta x_{dyne} + A_k^d \delta x_{kinE} + B u \\ \frac{d}{dt} \delta x_{kinE} = \delta x_{dyne} + A_k^k \delta x_{kinE} \end{cases} \quad (40)$$

where  $\delta x_{dyne} = (\delta v_E \ \delta \omega_E)^T \in \mathbb{R}^6$  denotes the 6 dynamic error states (velocity errors) and  $\delta x_{kinE} = (\delta p_E \ \delta \lambda_E)^T \in \mathbb{R}^6$  are the 6-dimensional kinematic states. The error dynamics system matrices  $A_d^d$  and  $A_k^k$  are defined as

$$A_d^d = \begin{bmatrix} A_v^v & A_v^w \\ A_w^v & A_w^w \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad A_k^k = \begin{bmatrix} 0 & A_v^\eta \\ 0 & A_w^\eta \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad (41)$$

respectively. The matrix  $A_d^d$  indicates the mutual interaction between the linear velocities and angular velocities, while the matrix  $A_k^k$  represents the influence of the kinematic states on the dynamic states. The kinematic error system matrix  $A_k^k$  is defined as

$$A_k^k = \begin{bmatrix} -S(\omega_T) & -S(v_T) \\ 0 & -S(\omega_T) \end{bmatrix} \in \mathbb{R}^{6 \times 6}. \quad (42)$$

The input matrix is given as  $B = (B_v \ B_w)^T \in \mathbb{R}^{(n_t+n_f) \times 6}$ . From the previous analysis, we come to the conclusion that the linearized error dynamic system, see Fig. 5, is a bijective function of trim trajectories  $p_T : (\|v_T\|, \psi_T, \gamma_T)^T$ . This property is of significance, since we can adopt the methods for MIMO linear systems to handle the originally nonlinear and strongly coupled system. Suppose the robot should track  $m$  trim trajectories, which can be combined as *motion primitives* to form a complex path. By implementing the aforementioned nonlinear

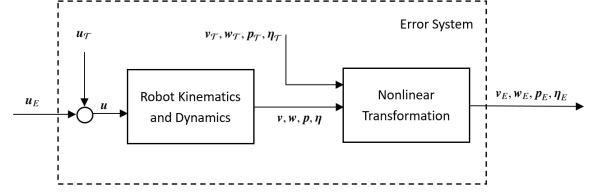


Fig. 5. Error System

transformation and linearizing at the trim equilibrium points, we will obtain  $m$  linearized error dynamic systems

$$\dot{x}_{E,j} = A_{E,j} x_{E,j} + B_{E,j} u_{E,j}, \quad \text{for } j = 1, \dots, m, j \quad (43)$$

representing the index for the trim trajectory, where  $x_{E,j} = (v_{E,j} \ w_{E,j} \ p_{E,j} \ \eta_{E,j})^T$ , the error dynamics system matrix

$$A_{E,j} = \begin{bmatrix} A_{d,j}^d & A_{k,j}^d \\ I_{6 \times 6} & A_{d,j}^d \end{bmatrix} \in \mathbb{R}^{12 \times 12} \quad (44)$$

and the error dynamics input matrix

$$B_{E,j} = \begin{bmatrix} B_{v,j} \\ B_{w,j} \end{bmatrix} \in \mathbb{R}^{12 \times (n_t+n_f)}. \quad (45)$$

For the complete trim path, we can formulate the error system along the  $m$  trim trajectories as a switched system

$$\Sigma_{S, A_E, B_E} : \dot{x}_E(t) = A_E(t) x_E(t) + B_E(t) u_E(t), \quad (46)$$

where  $A_E(t) \in \{A_{E,1}, \dots, A_{E,m}\}$  and  $B_E(t) \in \{B_{E,1}, \dots, B_{E,m}\}$ .

### 3.5 Switched LQR

For each linearized trim segment, we design a linear-quadratic regulator with weighting matrix  $Q_{E,j}$  and  $R_{E,j}$ , for  $j = 1, \dots, m$ , to minimize the cost function for the  $j$ -th trim trajectory

$$J = \int_0^\infty [x_{E,j}^T Q_{E,j} x_{E,j} + u_{E,j}^T R_{E,j} u_{E,j}] dt \quad (47)$$

subject to  $x_{E,j}(0) = x_{E,j,0}$ ,  $Q_{E,j} \geq 0$ ,  $Q_{E,j}^T = Q_{E,j}$ ,  $R_{E,j} > 0$  and  $R_{E,j}^T = R_{E,j}$ . Assuming that  $(A_{E,j}, B_{E,j})$  is controllable, the optimal solution to minimize the cost function is  $u_{E,j} = -K_j x_{E,j}$ , where  $K_j = R_{E,j}^{-1} B_{E,j}^T S_{E,j}$  is a constant linear state feedback gain. The matrix  $S_{E,j} \geq 0$  which is the solution of the continuous Algebraic Riccati Equation (ARE):

$$Q_{E,j} + A_{E,j}^T S_{E,j} + S_{E,j} A_{E,j} - S_{E,j} B_{E,j} R_{E,j}^{-1} B_{E,j}^T S_{E,j} = 0. \quad (48)$$

The input vector of the actuators is  $u_j = u_{E,j} + u_{T_j} = u_{T_j} - K_j x_{E,j}$ . When we implement the corresponding LQR state feedback controllers to all trim trajectory segments, we obtain  $\dot{x}_{E,m} = (A_{E,j} - R_{E,j} B_{E,j}^T S_{E,j}) x_{E,j}$ . If we denote the closed-loop system matrix as  $\bar{A}_{E,j} = A_{E,j} - R_{E,j} B_{E,j}^T S_{E,j}$ , we can represent the closed-loop error dynamics as a switched system

$$\Sigma_{S, \bar{A}_E} : \dot{x}_E = \bar{A}_E(t) x_E, \quad (49)$$

where  $\bar{A}_E(t) \in \{\bar{A}_{E,1}, \bar{A}_{E,2}, \dots, \bar{A}_{E,m}\}$ . An LQR can ensure the stability of each trim trajectory segment, however it can not ensure the stability of the whole trajectory. The stability for arbitrary switching can be proven by means of common quadratic lyapunov functions (CQLFs), see Shorten et al. (2007).

Recall that

$$V_{E,j}(x_{E,j}) = x_{E,j}^T P_{E,j} x_{E,j} \quad (50)$$

is a quadratic Lyapunov function (QLF) for the LTI system

$$\dot{x}_{E,j} = \bar{A}_{E,j}(t) x_{E,j}(t), \quad (51)$$

where  $P_{E,j}$  is symmetric and positive definite satisfying  $P_{E,j} \bar{A}_{E,j} + \bar{A}_{E,j}^T P_{E,j}$  being negative definite. The existence of CQLF



$V(\mathbf{x}_E) = \mathbf{x}_E^T \mathcal{P}_E \mathbf{x}_E$  is equivalent to determine whether such a matrix  $\mathcal{P}_E$  exists for a given group of Hurwitz matrices  $\{\bar{A}_{E,1}, \bar{A}_{E,2}, \dots, \bar{A}_{E,m}\}$  such that

$$\bar{A}_{E,m}^T \mathcal{P}_E + \mathcal{P}_E \bar{A}_{E,m} < 0. \quad (52)$$

This is a system of linear matrix inequalities (LMIs), which is said to be feasible if a solution  $\mathcal{P}_E$  exists. Otherwise, the LMIs are infeasible. Thus, determining whether or not the switched system  $\Sigma_{S, \bar{A}_E}$  has a CQLF amounts to checking the feasibility of a system of LMIs. We can convert these to the nonstrict LMIs  $\bar{A}_{E,m}^T \mathcal{P}_E + \mathcal{P}_E \bar{A}_{E,m} \leq I$  and  $\mathcal{P}_E \geq I$ . This can be formulated as semidefinite programming, which can be solved for example with the CVX toolbox by Grant et al. (2008, 2014), to check whether a matrix  $\mathcal{P}_E$  satisfying the aforementioned LMIs can be found.

#### 4. NUMERICAL SIMULATION

In order to verify the proposed switched LQR for the modular model, we apply rapid prototyping in MATLAB/SIMULINK. The robot geometric configuration parameters are adapted from the *Snookie* robot configuration described in Vollmayr et al. (2014), which serves as a baseline. The robot possesses four horizontal thrusters, two vertical thrusters, two horizontal fins and two vertical fins as actuators, see Fig 6. The thrust force is constrained to a range of  $[-35N, 40N]$ , and the deflection angle is constrained within  $[-20^\circ, 20^\circ]$ . The desired trim paths consist of seven segments whose parameters are specified in Table 1. The robot should track the seven trajectories in 150 seconds. Note that the yaw rate  $\dot{\phi}_T$  can not be set to zero due to the singularity problem of (14). Instead, we use a very small value ( $10^{-6}$ ) to tackle this problem. From the specifications and geometric configuration parameters, we can derive the linearized error systems for each trim trajectory segment:  $A_{E,1}, \dots, A_{E,7}$  and  $B_{E,1}, \dots, B_{E,7}$ . The initial state error  $\mathbf{x}_{E,1}$  is set to be  $\mathbf{0}$ . The drag coefficient  $c_D$  and lift coefficient  $c_L$  are set to be equal 0.1 and 0.001, respectively.

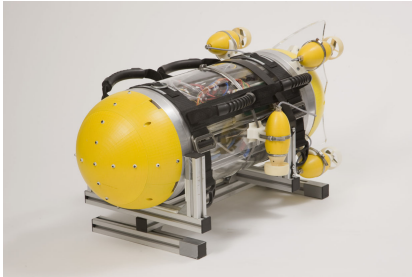


Fig. 6. *Snookie* AUV

Table 1. Trim trajectories specifications

Segment	Time(s)	$\ \mathbf{v}_T\ $ (m/s)	$\dot{\psi}_T$ (rad/s)	$\gamma_T$ (rad)
1	0-20	0.5	0	0
2	20-30	0.5	0.0803	0
3	30-40	0.5	0.0803	0.2
4	40-50	0.5	0	0.2
5	50-60	0.5	-0.0803	0.2
6	60-70	0.5	-0.0803	0
7	70-150	0.5	0	0

The state cost matrices for the individual trim segments are:

$$Q_{1/2/4/5} = \text{diag}([10^3, 10^3, 10^3, 300, 300, 300, 1, 1, 1, 1, 1, 1])$$

$$Q_3 = \text{diag}([10^3, 10^3, 10^3, 1, 1, 1, 1, 1, 1, 1, 1, 1])$$

$$Q_{6/7} = \text{diag}([10^3, 10^3, 10^3, 10^3, 10^3, 10^3, 1, 1, 1, 1, 1, 1]),$$

The input weighting matrices are:

$$\mathcal{R}_{2/3/4/5} = \text{diag}([10^{-4}, \dots, 10^{-4}, 10^{-3}, 10^{-3}, 10^{-3}, 10^{-3}])$$

$$\mathcal{R}_{1/6} = \text{diag}([10^{-4}, \dots, 10^{-4}]) \in \mathbb{R}^{12 \times 10}$$

$$\mathcal{R}_7 = \text{diag}([10^{-5}, \dots, 10^{-5}, 10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}])$$

Using these values, we can design an LQR for each linearized error system and obtain state feed back gains  $\mathcal{K}_1, \dots, \mathcal{K}_7$ .

In order to ensure the global stability of the formulated switched system under implementation of the LQR with the selected weighting matrices, we check the existence of the common symmetric matrix  $\mathcal{P}_E$  by solving the following optimization problem:

$$\min_{\mathcal{P}_E \in \mathbb{R}^{12 \times 12}} \text{tr}(\mathcal{P}_E), \quad (53)$$

subject to  $\bar{A}_{E,j}^T \mathcal{P}_E + \mathcal{P}_E \bar{A}_{E,j} \leq I_{12 \times 12}$ ,  $\mathcal{P}_E \geq I_{12 \times 12}$ , where  $\bar{A}_{E,j} = A_{E,j} - \mathcal{K}_j$ ,  $j = 1, \dots, 7$ . Using the semidefinite programming in CVX (Grant et al. (2008, 2014)), we could obtain the optimization result. For the current values of state cost matrices and input cost matrices, the objective function  $\text{tr}(\mathcal{P}_E)$  has a feasible solution of 1891.49, such that the corresponding positive definite symmetric matrix  $\mathcal{P}_E$  can be used to prove the global stability of the linear switched error system.

The reference trim trajectories and the tracked path of the underwater robot during the simulation are illustrated in Fig. 7 and the corresponding error states  $\mathbf{p}_E, \boldsymbol{\eta}_E, \mathbf{v}_E, \mathbf{w}_E$  are shown in Fig 8. If there is no actuator constraint, the robot can realize perfect tracking with the switched LQR. For practical implementation, the control inputs are constrained by the capabilities of the actuators, as illustrated in Fig. 9. For tracking the first and the second trim segment along which the flight angle equals zero, the robot can nearly realize perfect tracking. When it moves downwards with flight angle 0.2 rad, all error states rise and oscillate obviously, especially the pitch error  $\theta_E$ . Another increase of state errors happens at the moment when the flight path angle  $\gamma_T$  falls from 0.2 to 0. The changing of flight angle results in huge error and strong oscillations of the pitch error. At 70 seconds, the yaw rate becomes zero suddenly, with a large roll velocity error  $\mathbf{p}_E$  appearing consequently. Although we implement an LQR for each trim trajectory segment, the controller can not drive the error states to zero before switching to the next trim segment. The remaining errors are the initial values for the next error dynamic system. In order to show the convergence performance of the error states under the implemented LQR, we extend the duration of the last trim segment to 80 seconds. From Fig. 8, we can see that the error states converge gradually to zero.

In summary, each trim path switching leads to a deviation of the real trajectory from the desired one. A switched LQR control strategy with suitably chosen state and input weighting matrices can ensure the global stability and drive the robot to track the reference trajectory.

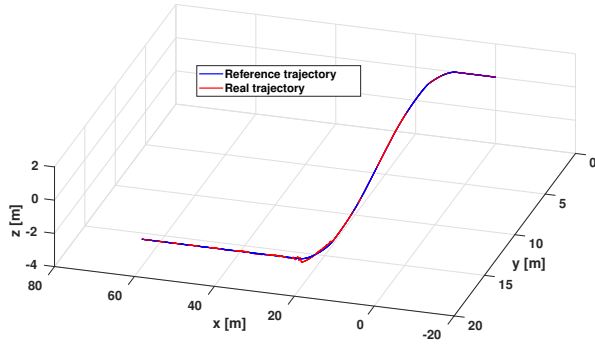


Fig. 7. Tracking result

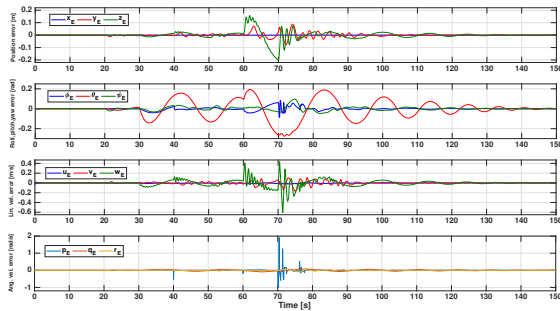


Fig. 8. Error states

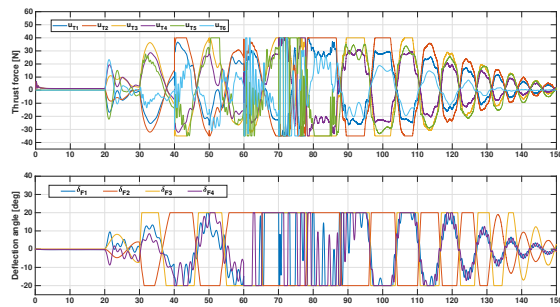


Fig. 9. Thrust force and deflection angle

## 5. CONCLUSIONS

In preparation for formulating the geometric design procedure as an optimization algorithm, in this paper we introduce a new modular modeling approach for (autonomous) underwater vehicles. The focus is on the identification of the geometric decision variables and their couplings both for the overall kinematic and kinetic design of the robot as well as the control allocation. In addition, we propose a method to formulate the robot dynamics as a linear MIMO switched system based on the task-determined trim trajectories and synthesize subsequently a switched LQR controller, which can be checked for global stability. This enables a novel fully automatic task-dependent co-design of an AUV and its controller. The simulation results demonstrate that, in the unconstrained actuator case, the proposed model and control strategy can be successfully applied to perfectly track a set of predefined trim trajectories. However, the results also show that since LQR does not explicitly consider actuator constraints, the tracking performance under

more realistic conditions are suboptimal. To take the actuator constraints into consideration, a switched Model Predictive Controller (MPC) or a Linear parameter-varying (LPV) control approach is suggested for future work.

## REFERENCES

- Antonelli, G., Underwater Robots. Berlin Heidelberg: Springer, 2003.
- Bestaoui, Y. and Hima, S., Modelling and trajectory generation of lighter-than-air aerial robots-invited paper, in Robot Motion and Control 2007, 3-28. London: Springer, 2007.
- Chen, Y., Modular modeling and control for autonomous underwater vehicle (auv). Masters thesis, National University of Singapore, Dept. of Mechanical Engineering, 2007.
- Cunha, R. and Silvestre, C., A 3D Path-Following Velocity-Tracking Controller for Autonomous Vehicles, in 16th IFAC World Congress, Praha, Czech Republic, 2005, July.
- Du, T., Schulz, A., Zhu, B., Bickel, B., and Matusik, W., Computational multicopter design. ACM Trans. Graph., 35(6):227:1-227:10, November 2016.
- Grant, M. and Boyd, S., Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, pages 95-110. Springer-Verlag Limited, 2008.
- Grant, M. and Boyd, S., CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- Kaminer, I. and Pascoal, A. and Hallberg, E. and Silvestre, C., Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control, AIAA Journal of Guidance, Control, and Dynamics, 21(1): 29-38, 1998.
- Nahon, M., A simplified dynamics model for autonomous underwater vehicles, Proceedings of Symposium on Autonomous Underwater Vehicle Technology, Monterey, CA, 1996, pp. 373-379. doi: 10.1109/AUV.1996.532437
- Sebbane, Y. B., Planning and Decision Making for Aerial Robots. Berlin Heidelberg: Springer, 2014.
- Shorten, R., Wirth, F., Mason, O., Wulff, K., and King, C., Stability Criteria for Switched and Hybrid Systems. SIAM Review, 49(4):545-592, 2007.
- Shuzhe, C., Soon, H. G., Hong, E. Y. and Chitre, M., Modular modeling of autonomous underwater vehicle, OCEANS'11 MTS/IEEE KONA, Waikoloa, HI, 2011, pp. 1-6. doi: 10.23919/OCEANS.2011.6106921
- Silvestre, C., Pascoal, A., and Kaminer, I., On the design of gain-scheduled trajectory tracking controllers. International Journal of Robust and Nonlinear Control, 12(9):797-839, 2002.
- Tang, S., Modeling and simulation of the autonomous underwater vehicle, autolycus. Master's thesis, Massachusetts Institute of Technology, Dept. of Ocean Engineering, 1999.
- Vollmayr, A.N., Sosnowski, S., Hirche, S., and van Hemmen, J.L., Snookie: an autonomous underwater vehicle with artificial lateral line system. In Horst Bleckmann, Joachim Mogdans, and Sheryl L. Coombs, editors, Flow Sensing in Air and Water, chapter 20, pages 521-562. Springer Berlin Heidelberg, Berlin, Heidelberg, viii edition, 2014.
- Yoerger, v, Cooke, J. G., and Slotine, J. J. E., The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. IEEE Journal of Oceanic Engineering, 15(3):167-178, July 1990.