Supplementary Documentation – Verifying UUV Pipe Following Safety Using Verisig

In this case study, there is an Unmanned Underwater Vehicle (UUV) equipped with two side-scan sonars (one on each side) that is tasked with following an undersea pipeline that is somewhere on the seafloor. While the starting location of the pipeline is known, the UUV must use its sonar capabilities to follow the pipeline, while maintaining appropriate distance and relative heading to maximize the resolution of the sonar image. Refer to the main documentation at the root of the directory for further information regarding the use of Verisig.

1 Full System Description

In the original design, the system uses two learning-enabled components (LECs), one that performs perception and one that performs control. The perception LEC takes the side-scan sonar as input and, after some postprocessing, outputs a relative distance and bearing to the pipe. The control LEC takes the output of the perception LEC and outputs desired heading, speed and depth (HSD), which are then sent to a stack of PID controllers that send raw control commands to the fins and the thruster.

2 Abstracted System Description

In this case study, our goal is to verify the safety of the control LEC, assuming the perception LEC is replaced with ground truth information. Specifically, we focus on the following system model.

2.1 Plant Model

We will focus on a scenario with constant depth control, i.e., we will consider a system with two-dimensional position. For this system, we have performed system identification, as described in Section 4. We assume the plant dynamics are given as a 4-th order linear time-invariant system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k, \end{aligned}$$

where $A \in \mathbb{R}^{4\times4}$, $B \in \mathbb{R}^{4\times3}$, $C \in \mathbb{R}^{3\times4}$, $D \in \mathbb{R}^{3\times3}$ are identified using standard system identification techniques as described below. Note that the $x \in \mathbb{R}^4$ states are purely digital and have no physical interpretation per se. Also note that the model implicitly includes the PID controllers as well.

The controls $u \in \mathbb{R}^3$ are the outputs of the control LEC and correspond to desired heading (in radians), speed, and depth, respectively. In this scenario, we assume that the depth is held constant at 45m, and speed is held constant at 1 m/s. Thus, the control LEC only outputs one-dimensional heading (in the $x - y$ plane).

The measurements $y_k := [y_k^0, y_k^1, y_k^2] \in \mathbb{R}^3$ are: heading (in radians), speed, and depth. Again, note that the heading is one-dimensional in the $x - y$ plane.

Finally, note that we also need to know the UUV's absolute position (which is not a part of the identified model above). We calculate that using the following dynamical system:

$$\begin{bmatrix} y_k^3 \\ y_k^4 \end{bmatrix} = \begin{bmatrix} y_{k-1}^3 + t_s * y_k^1 * cos(y_k^0) \\ y_{k-1}^4 + t_s * y_k^1 * sin(y_k^0) \end{bmatrix},$$

where y_k^3 and y_k^4 are x and y position, respectively, and t_s is the control sampling rate. We assume $t_s = 1s$.

2.2 Control LEC

The control LEC is a tanh-based neural network, with 2 hidden layers with 32 neurons per layer, and an output layer with one neuron. The control LEC takes three inputs, denoted by $y_k^{i1}, y_k^{i2}, y_k^{i3}$: pipe heading relative to the UUV (in radians), distance to the pipe on the starboard side, and distance to the pipe on the port side. The control LEC outputs one value, the desired heading relative to the UUV's current heading; the output is constrained to at most 5 degrees in either direction. Note that if there distance to pipe is negative on the side of the UUV facing away from the pipe. We emphasize again that the inputs to the control LEC are ground truth (explained in the next subsection), and are not the outputs of the perception LEC as would be the case in the full system.

2.3 Pipe layout

We assume the pipe is straight in this scenario. Without loss of generality, we assume the pipe starts at the origin and coincides with the x-axis. The vehicle is assumed to be heading north initially, i.e., in the positive x direction. Given this assumption, the inputs to the control LEC could be described as:

$$\begin{bmatrix} y_k^{i1} \\ y_k^{i2} \\ y_k^{i3} \end{bmatrix} = \begin{bmatrix} -y_k^0 \\ -y_k^4/cos(y_k^0) \\ y_k^4/cos(y_k^0) \end{bmatrix}.$$

2.4 Composed System description

For completeness, this section summarizes the above parts and provides a mathematical description of the entire closed-loop system (with some constants replaced with their actual values for

simplicity).

$$\begin{bmatrix} x_{k+1}^0 \\ x_{k+1}^1 \\ x_{k+1}^2 \\ x_{k+1}^3 \end{bmatrix} = Ax_k + Bu_k$$

$$\begin{bmatrix} y_k^0 \\ y_k^1 \\ y_k^2 \end{bmatrix} = Cx_k + Du_k$$

$$\begin{bmatrix} y_k^3 \\ y_k^4 \end{bmatrix} = \begin{bmatrix} y_{k-1}^3 + y_k^1 * cos(y_k^0) \\ y_{k-1}^4 + y_k^1 * sin(y_k^0) \end{bmatrix}$$

$$\begin{bmatrix} y_k^{i1} \\ y_k^{i2} \\ y_k^{i3} \end{bmatrix} = \begin{bmatrix} -y_k^0 \\ -y_k^4/cos(y_k^0) \\ y_k^4/cos(y_k^0) \end{bmatrix}$$

$$\begin{bmatrix} u_k^0 \\ u_k^1 \\ u_k^2 \end{bmatrix} = \begin{bmatrix} y_k^0 + 0.0872665 * LEC(y_k^{i1}, y_k^{i2}, y_k^{i3}) \\ 1 \\ 45 \end{bmatrix},$$

where 0.0872665 is 5 degrees in radians. Note that since the LEC's output neuron has a tanh activation, the multiplication ensures that the output is bounded between -5 and 5 degrees (in radians).

3 Verification Problem

Given the abstracted system description in the previous section, the problem is to verify that the UUV always stays within 10 to 50 meters from the pipe. The UUV starts from an initial range of $y_0^4 \in [25, 35]$, i.e., between 25 and 35 meters from the pipe. Furthermore, the UUV starts with an initial speed and heading of 0, and an initial depth of 45.

4 System Identification

The model was identified from the ROS-based UUV simulator using standard system identification tools in Matlab. The identified model includes the interaction of the PID controllers that are present on the UUV. It is possible te resample the model for different control sampling rates, but we found the 1Hz model to be easiest to verify.

Note that the linear model is only accurate within the envelope of the scenario described in the Section 2. Specifically, the model was identified assuming constant depth a maximal control output of -5 to 5 degrees.

5 LEC Training

A neural network controller was trained using reinforcement learning. The controller takes three inputs: pipe heading relative to the UUV (in radians), distance to the pipe on the starboard side, and distance to the pipe on the port side. It produces a single output, the desired heading relative to the UUV's current heading; the output is constrained to at most 5 degrees in either direction. The reward function was chosen as to force to controller to keep a relative pipeline heading of 0

and a distance to pipeline of 30 meters. We used a twin-delayed deep deterministic policy gradient (TD3) algorithm to train the LEC. Note that the control LEC in the ROS simulator could not be used as it is not a standard feed-forward neural network (the output layer outputs the parameters of a beta distribution, which is then sampled to choose the specific control action). The reinforcement learning was conducted in a simplified, OpenAI Gym-like environment[1]. The training environment is included in the *training* subdirectory.

6 SpaceEx Model

In order to use Verisig to verify the problem, we need to encode the problem in the SpaceEx editor (please consult the main documentation for detailed instructions). The UUV SpaceEx model closely follows the description in Section 2.4. Note that the k subscripts are removed from the SpaceEx model. Note that Flow* (the tool which Verisig is based on) does not support non-polynomial functions in discrete resets, such as the trig functions and division. That is why Verisig adds limited functionality for these functions – they can be performed one at a time, through using corresponding hybrid system mode names (please consult the main documentation for more details). This explains why the SpaceEx model has multiple modes. Finally, note that we use the `_f` states to capture the inputs and outputs to the LEC (please consult the main documentation for more details).

7 Verification

As noted above, the verification problem is to verify that the UUV will stay within 10 to 50 meters from the pipe, as started between 25 and 35 meters from the pipe. The initial condition had to be split up into multiple regions in order to keep the uncertainty small. Specifically, the initial condition [25, 29] was subdivided into subintervals of size 0.02 (e.g., [25, 25.02], [25.02, 25.04], etc.), wheres the region [29, 35] was subdivided into subintervals of size 0.01. The script *multi_runner.py* can be used to run a batch of verification instances in parallel. It is currently set up to run 50 instances between [29, 29.5]: each instance takes about an hour on a moderately-powered laptop. The multi-runner will spawn processes equal to half the number of CPUs.

[1] https://gym.openai.com/