



EIVA



SCANFISH FLIGHT V1.5

AUTOMATIC 3D CONTROL USERGUIDE

Last update: 19/04/2023
Version: 2.0

Contents

1	Introduction	3
1.1	Formatting conventions	3
2	Enabling the automatic 3D control	3
2.1	Setting up the importer	4
3	Engaging and disengaging the autopilot	7
4	AutoPilot view	8
4.1	AutoPilot view – Disabled	9
4.2	AutoPilot view – Enabled	10
4.2.1	Segment regulation	11
4.2.2	Parallel PID regulation	11
5	Adjusting the autopilot settings	12
5.1	Segment regulation settings	12
5.2	PID regulation settings	13
6	Autopilot logic	13
6.1	Segment logic	13
6.2	Parallel PID logic	15
6.2.1	Proportional gain	16
6.2.2	Integral gain	16
6.2.3	Derivative gain	16

1 Introduction

This document is a quick description of how to set up, enable and use the automatic 3D control system for the ScanFish Flight version 1.5.

1.1 Formatting conventions

Items formatted in **bold** are properties, buttons, or other elements in the ScanFish Flight software, when setting up the ScanFish automatic 3D control.

2 Enabling the automatic 3D control

The 3D control system works by using NMEA XTE, measured cross track error data, to navigate the fish to a desired location. To access the 3D system controls an importer must be set up first.

The Autopilot Importer receives NMEA XTE data indicating how far the ScanFish is from reaching its target run line.

2.1 Setting up the importer

Go to **Settings** and select **Options**.

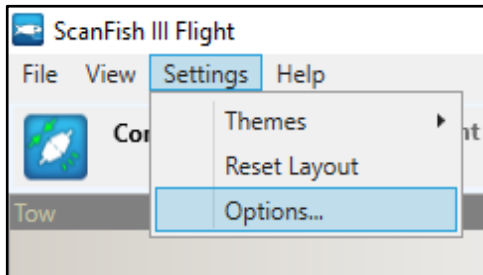


Figure 1 Finding the options menu

Open the **Importers** submenu and select **Autopilot Importer** tab.

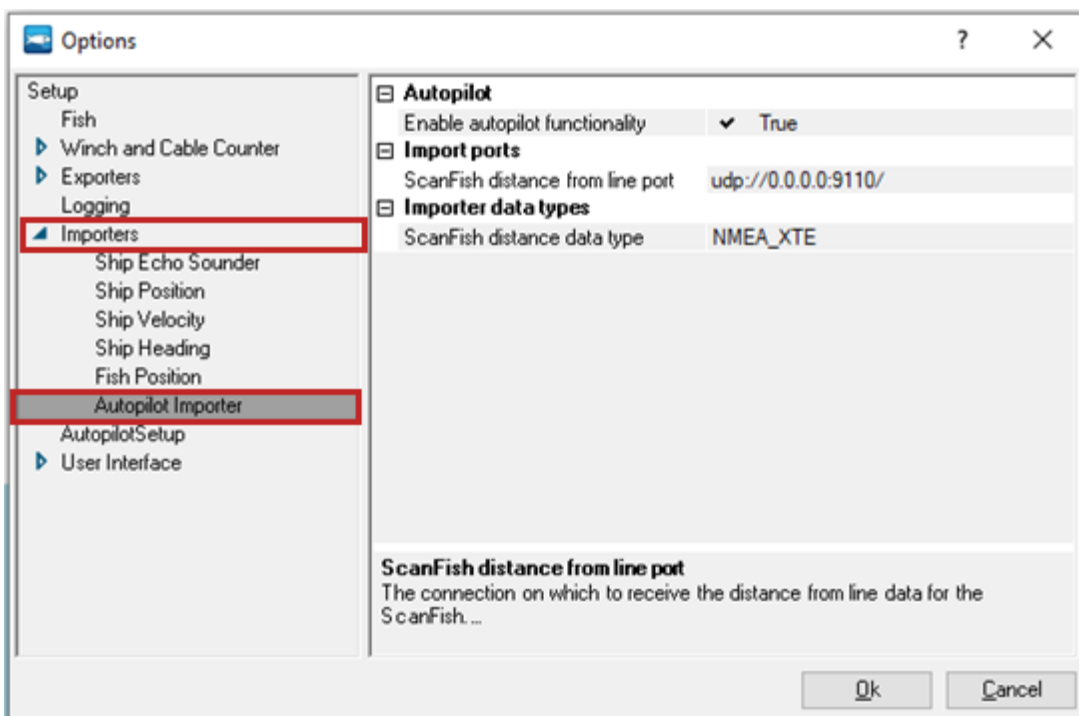


Figure 2 Autopilot Importer options menu

Set **Enable autopilot functionality** to **true**, the **ScanFish distance data type** to **NMEA_XTE** and set the **ScanFish distance from line port** to a **UDP** connection with the port number on which the data should arrive. You can do so by pressing the icon to the right with the three dots (shown below).

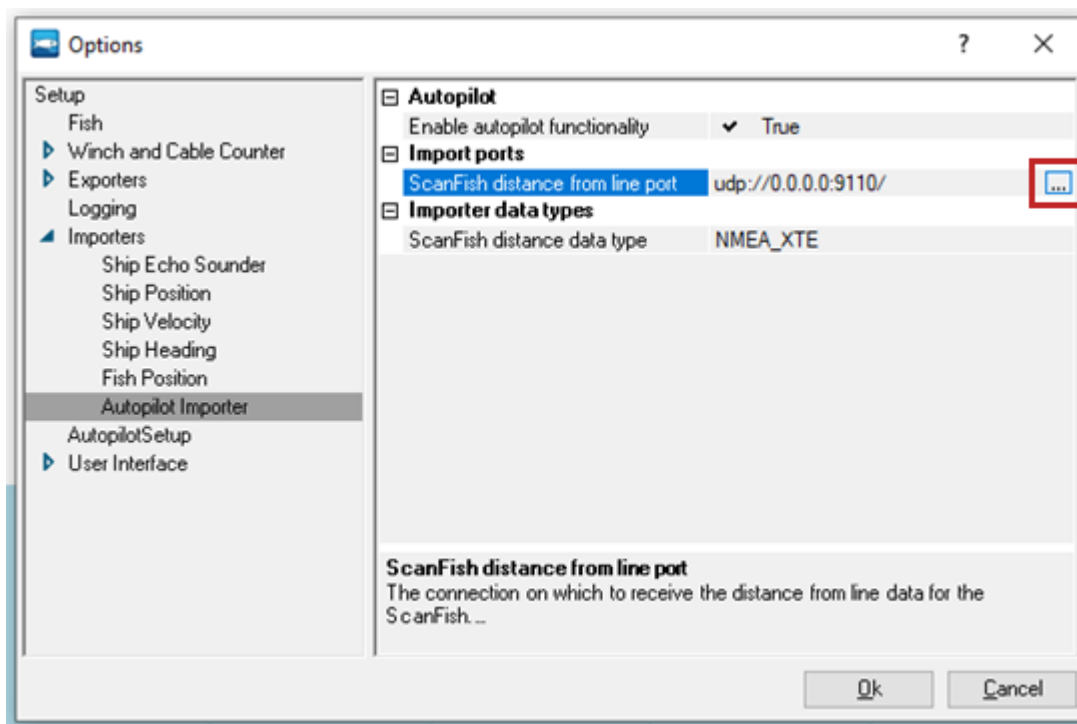


Figure 3 Setting the UDP port where data should arrive

We recommend setting the **I/O Type** to **UDP**. Make sure the system sending the cross-track error data is using the same **I/O Type**, otherwise no data will be received.

To check that you are receiving cross-track data, you can use the **Test** button. If you are receiving any input, it will be shown in the **I/O Output** box in the bottom of the window.

In the example below you can see that a NMEA XTE message was received.

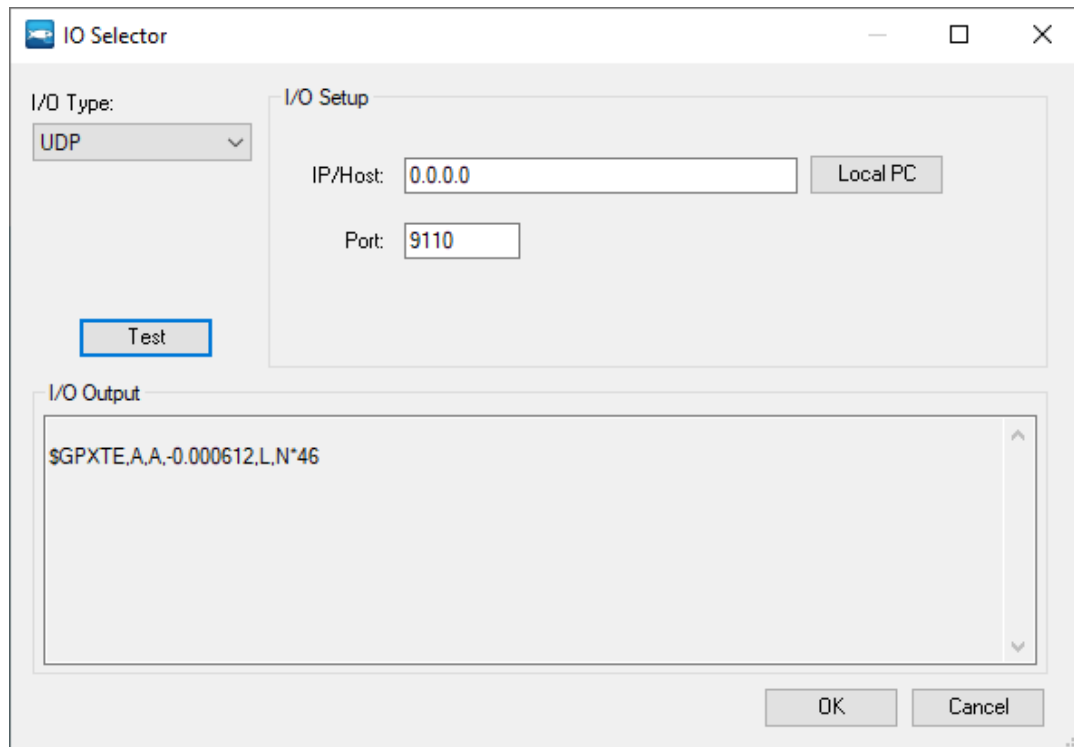


Figure 4 Setting and testing autopilot input

3 Engaging and disengaging the autopilot

Before the autopilot can be engaged you need to enable the **Autopilot Importer** and press the **Connect** button.

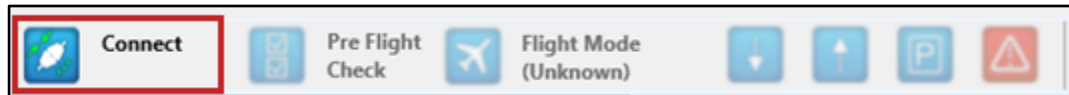


Figure 5 Connect button location

After connecting, the **Engage Autopilot** button will appear to right on the top bar.



Figure 6 Engage autopilot button location

Clicking the **Engage Autopilot** button the autopilot will start calculating target angles for the 3D rudders based on the input distance from the target line. If no input data is received no calculations will be done and the rudder targets will stay the same.

The **Engage Autopilot** button will the switch to a **Disable Autopilot** button. Pressing that will disable the autopilot and reset the 3D rudders to their home position.



Figure 7 Disengage autopilot button location

Note: Using the manual vertical flap handle in the **Tow panel** will disengage the autopilot if it is already engaged.

4 AutoPilot view

To control and monitor the autopilot, an **Autopilot view** is available. To open the autopilot view, select **View** in the top bar, then select **Dockable Panels** in the drop-down menu and choose **AutoPilotPanel** in the submenu.

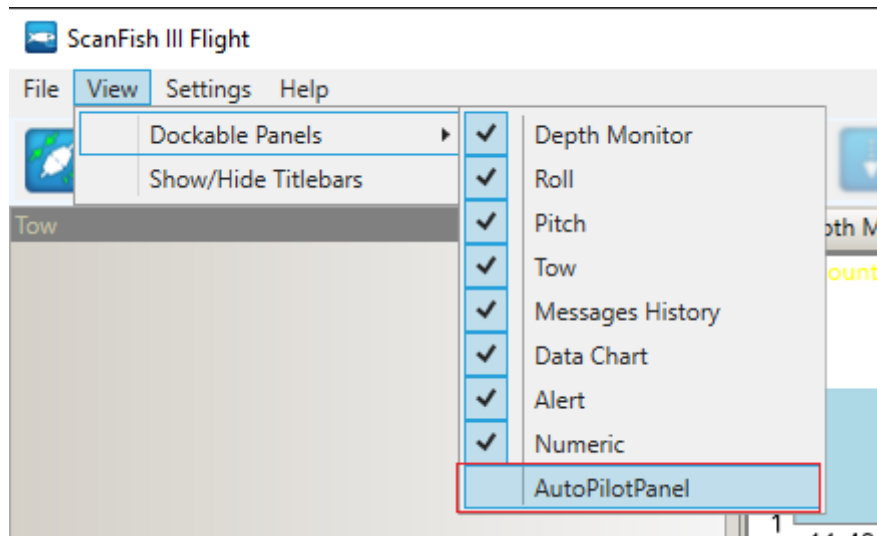


Figure 8 Opening the AutoPilot view

4.1 AutoPilot view – Disabled

When disabled, the **AutoPilot view** will only show the autopilot input value **Distance from target** and output value **Rudder target angle**. The autopilot input value will update even if the autopilot is disabled but no output will be sent to the ScanFish. This is to help show that the autopilot importer is working correctly.

In the **Disabled** mode, the flaps are locked in the zero position and will remain in this position until the flaps are moved manually. This is the default mode for the Autopilot.

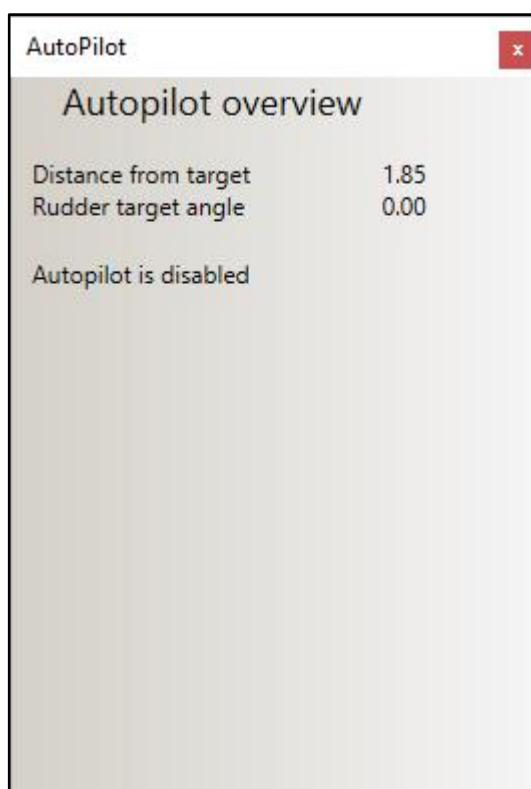


Figure 9 The disabled Autopilot view

4.2 AutoPilot view – Enabled

When enabled, the **AutoPilot view** changes to show all the information needed to monitor the autopilot system.

It is separated into three parts: **Autopilot overview**, **Segment regulation** and **Parallel PID regulation**.

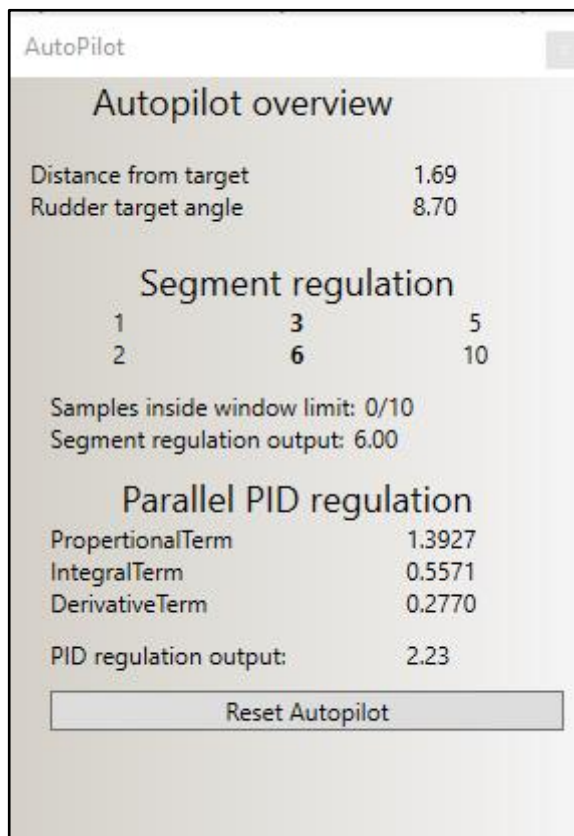


Figure 10 The AutoPilot view when enabled

The **Autopilot overview** shows the same as the when the autopilot is disabled. The only difference is that the **Rudder target angle** is now updated showing the final output angle calculated by autopilot logic sampling, and adds the **Segment regulation output** to the **PID regulation output**.

Rudder target angle = Segment regulation output + PID regulation output

4.2.1 Segment regulation

The **Segment regulation** has a simplified segment view based on the original **AutoPilot view**, along with a counter indicating how soon an update will happen to the segment calculations.

The **Segment regulation output** value shows the output for the segment part of the autopilot calculation.

Segment control											
Distance from target	-7.39	-4.95	-3.32	-2.23	-1.49	0.00	1.49	2.23	3.32	4.95	7.39
Flap offset value	-3.02	-2.29	-1.80	-1.47	-1.25	0.00	1.25	1.47	1.80	2.29	3.02

Figure 11 The segment control in the original autopilot view

4.2.2 Parallel PID regulation

The **Parallel PID regulation** shows the intermediate calculations of the proportional, integral and derivative parts of the PID regulation. Use these to monitor if they are weighted too high in the **Options** menu.

The **PID regulation output** value shows the output for the Parallel PID part of the autopilot calculation.

5 Adjusting the autopilot settings

To open the **Autopilot options** menu, click **Settings** and select **Options**. When the **Options** menu is open, select the **AutopilotSetup** tab.

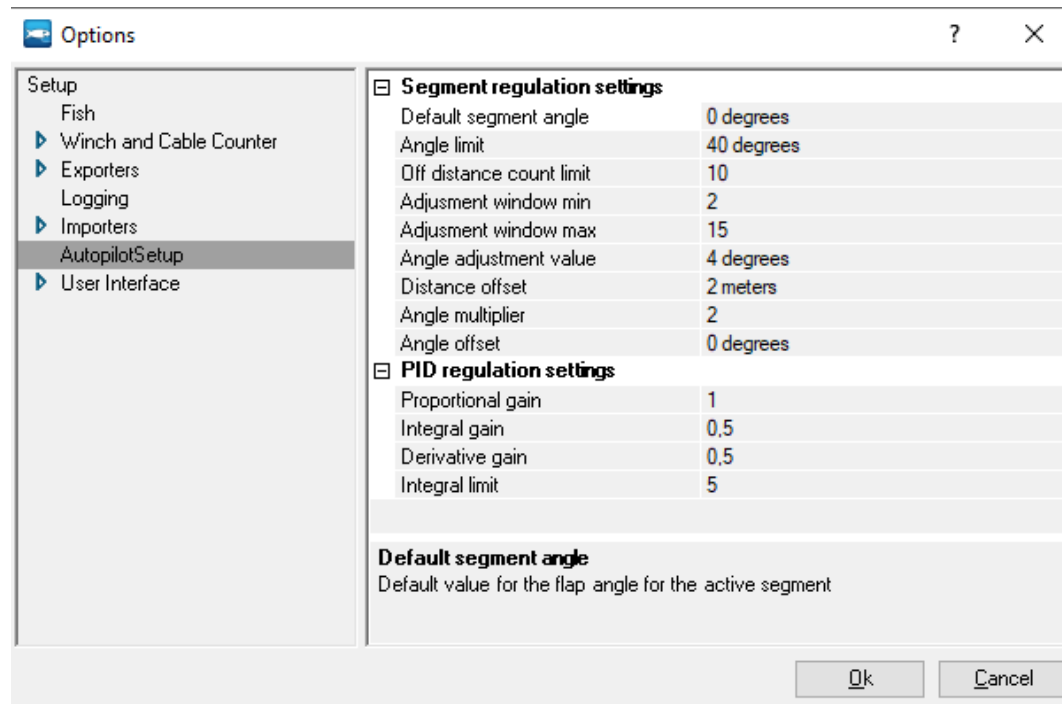


Figure 12 The AutopilotSetup menu

5.1 Segment regulation settings

- **Default segment angle**
Is the default value used at startup when the ScanFish is on the target line. It is recommended keeping this at zero
- **Angle limit**
Sets the maximum rudder angle that the regulation can set. Note that rudder motors have an angle limit around 40 degrees, so increasing the value further will have no effect
- **Off distance count limit**
How many samples should fall inside the Adjustment window before an adjustment should happen
- **Adjustment window min**
Sets the minimum distance for the adjustment window. It is recommended to not set this to zero

- **Adjustment window max**
Sets the maximum distance for the adjustment window.
- **Angle adjustment value**
Sets the angle value that is added or subtracted to the segments angle values
- **Distance offset**
Sets the distance between each segment
- **Angle multiplier**
Sets the angle value multiplier for each segment. If the multiplier is two and the segment distance value is 3 then the resulting angle will be 6

5.2 PID regulation settings

- **Proportional gain**
How aggressive the proportional part of the PID should be
- **Integral gain**
How aggressive the integral part of the PID should be
- **Derivative gain**
How aggressive the derivative part of the PID should be
- **Integral limit**
Sets a max limit to how big the integral value can be as the integral can otherwise run out of control

6 Autopilot logic

6.1 Segment logic

From the autopilot input a virtual target line is made. The ScanFish is then located to either the port or starboard side of that line (depending on the XTE input value). Each side of the target line is then split up into multiple segments, hence the Segment logic.

A segment is defined as a point on one of the two sides on the target line. Each segment has a distance and a flap angle associated with it so that if the ScanFish is located on a specific segment, its 3d rudders are set to the angle associated with the segment.

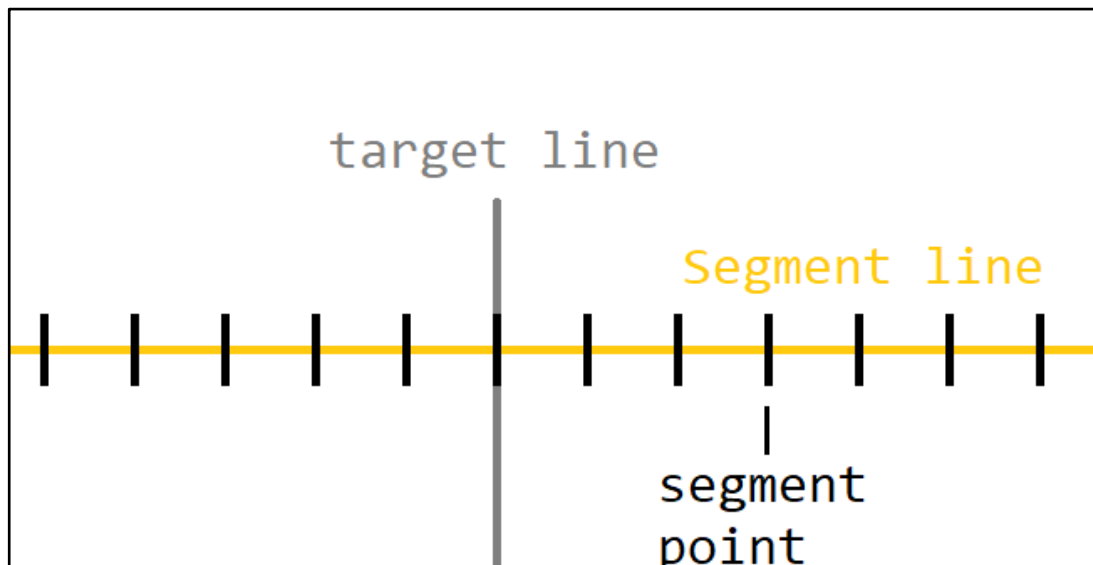


Figure 13 ScanFish sideways axis split up into segments

As an example, assume there is a segment for each metre on the sides and the corresponding target angle is the same as the segment distance. If the ScanFish is 3 metres from the target line, then the active segment will have a distance value of 3 and the target angle will also be 3. For each metre the ScanFish gets further away from the target line, the target angle for the 3D rudders will also increase.

The **Segment** section of the **AutoPilot window** is built up around the concept of the segment line. The upper row is the segment distance from the target line. The bottom row is the target angle for the 3D rudders.

The two values in the middle indicate the active segment. The value pairs on both sides indicate the segments closest to the active segment.

In the case shown below, the input values lie somewhere between 3 and 5 metres. As long as the distance from the line stays within 3 and 5 metres, the segment regulation output will be 6 degrees.

Segment regulation		
1	3	5
2	6	10
Samples inside window limit: 0/10		
Segment regulation output: 6.00		

Figure 14 - The Segment regulation part of the autopilot view

The **Segment regulation** has a built-in output adjuster. It increases or decreases the output angle for all segments based on the last update inside a certain window of values deemed too far away from the target line.

The **Adjustment window** is a distance range wherein an update counts as being far enough away from the target line to require an adjustment, but not too far away that too many updates would be made before the ScanFish has time to move towards the target line.

When enough samples have fallen inside the window, the segment angle values will be updated.

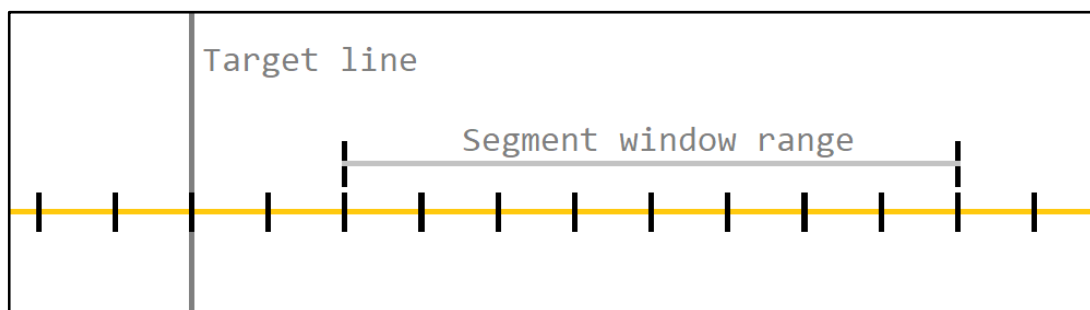


Figure 15 - Segment window range

6.2 Parallel PID logic

The **Parallel PID control** supplements the **Segment control**. Where the Segment control sets a static flap angle based on the ScanFish's distance to the target line, the Parallel PID control adds a dynamic flap movement centred on the static position set by the **Segment control**.

Parallel PID regulation	
ProportionalTerm	1.6983
IntegralTerm	0.7175
DerivativeTerm	0.2933
PID regulation output:	2.71

Figure 16 Overview of the Parallel PID regulation overview

- **ProportionalTerm**
The value is the proportional gain result and is the present position.
- **IntegralTerm**
The value is the integral gain result and is the past position history.
- **DerivativeTerm**
The value is the derivative gain result and is a prediction of the future position.

- **PID regulation output**
Shows the sum of the three terms above.

Output = P_term + I_term + D_term

The PID regulations can be fine-tuned using the **Autopilot setting** in the **Options** menu. There you can adjust the proportional, integral and derivative gain constants, as well as set a limit on the PID output and a damping mechanism for the integral gain.

6.2.1 Proportional gain

The **ProportionalTerm** shows the proportional gain constant which dictates how much the system should react to the latest incoming DOL data. It produces an output value that is proportional to the current distance to the target value. The proportional response is adjusted by multiplying the error by this constant.

A proportional gain value of **1** will result in a 1:1 ratio of the distance from target in metres to the flap angle. Increasing the value to **2**, will result in a 1:2 ratio, meaning if the distance from target is 2 metres, the result sent to the ScanFish will be 4.

Setting the proportional gain too high will result in a system that oscillates around the desired runline!

6.2.2 Integral gain

The contribution from the **IntegralTerm** is proportional to both the magnitude of the error and the duration of the error. The integral is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is multiplied by the integral gain constant and added to the output value.

It serves as a memory buffer for the past values sent to the flaps and helps accelerate the movement towards the desired position.

Note: An integral gain that is too high can result in the system overshooting the runline as the accumulated error values are larger than the proportional value!

6.2.3 Derivative gain

The derivative error is used to predict the systems behaviour, helping the system settle on the correct runline and overall system stability.