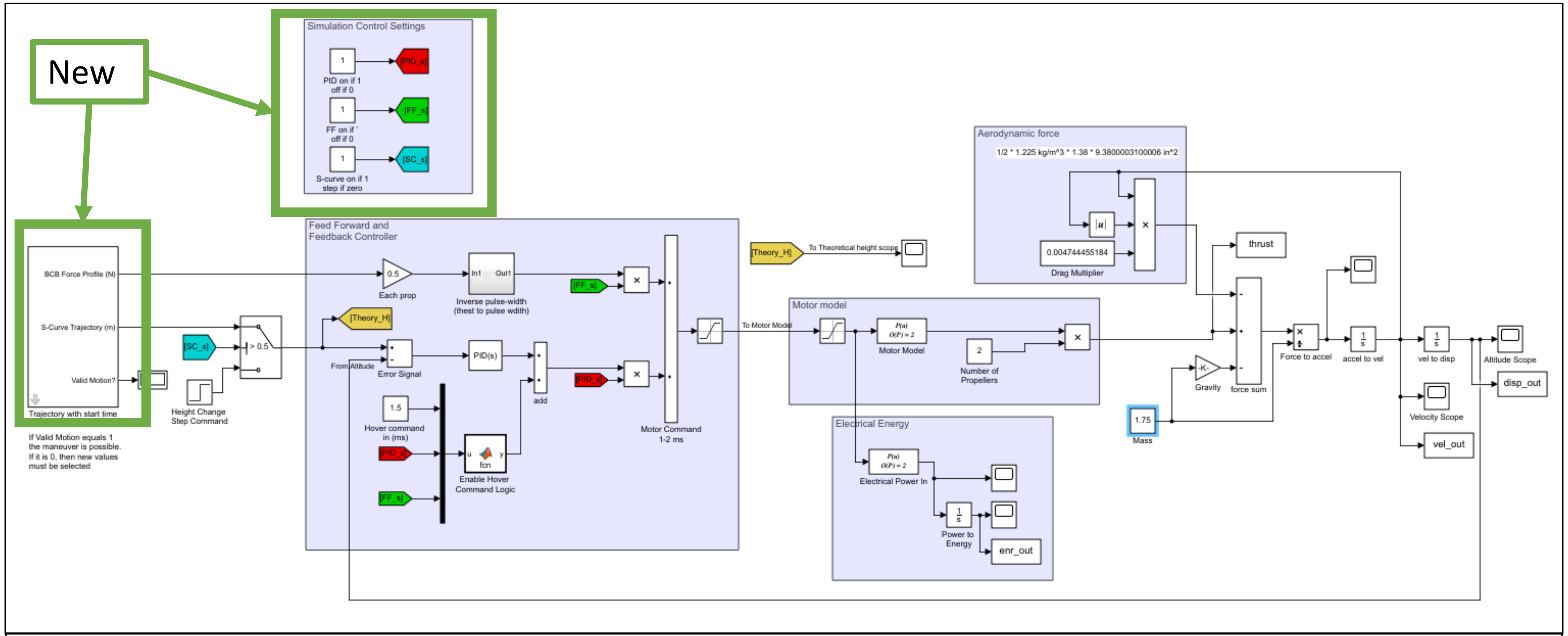
STARK INDUSTRIES

Kalman Filter Investigation

Overview

1. Simulink Model Update
2. LIDAR Data Variation
3. Kalman Filter Development
4. Kalman Filtered LIDAR Data
5. Kalman Filtered LIDAR Data Gains
6. Effect of Noise on PID and FF Control
7. Effect of Noise on Feedforward Control
8. Kalman Filter Implementation
9. Kalman Filtered Sensor Input 1
10. Kalman Filtered Sensor Input 2
11. Kalman Filtered Sensor Input 3
12. Velocity Measurement Exploration

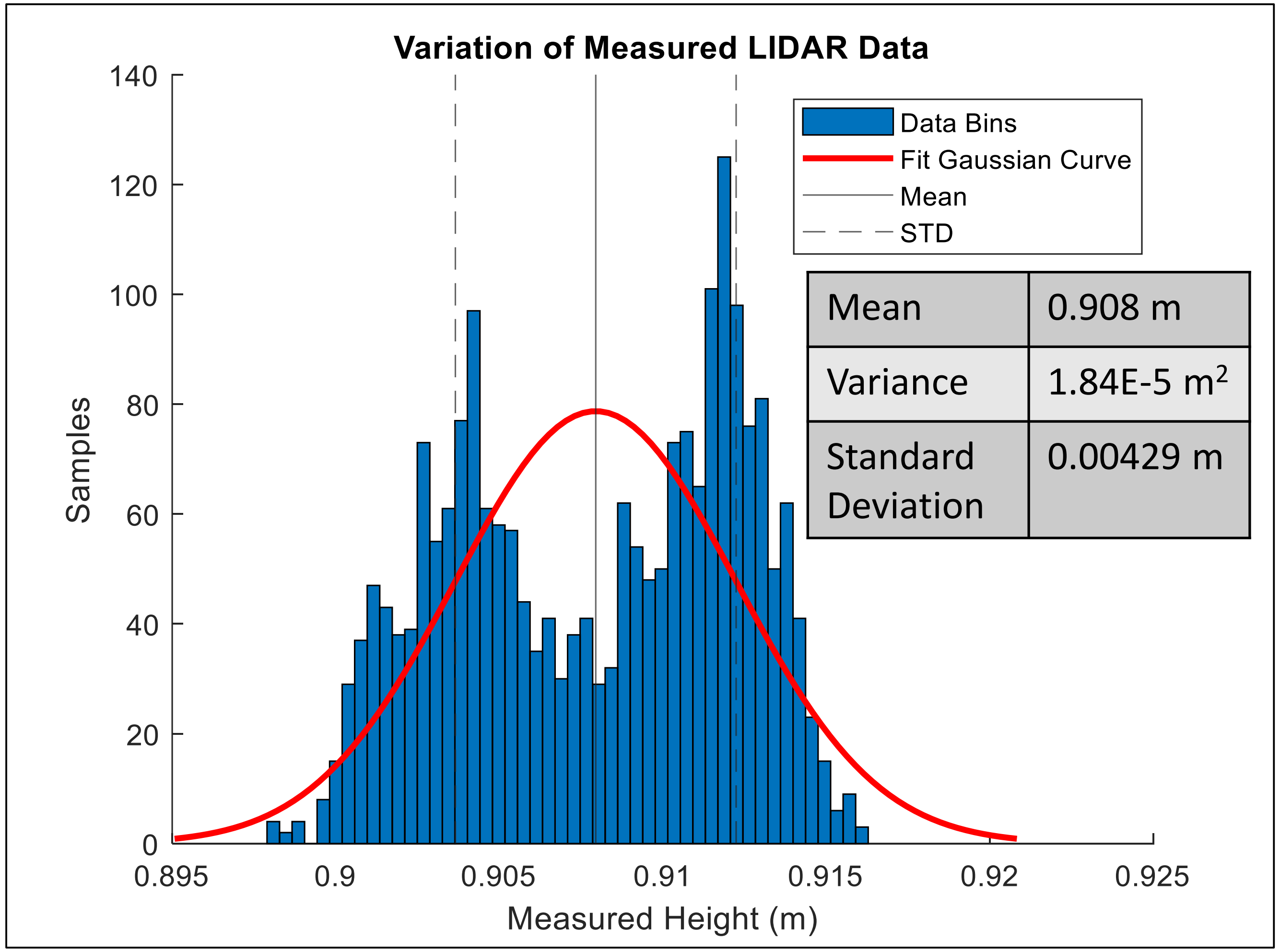
1. Simulink Model Update



The update Simulink model features a new bang-coast-bang block that is improved over the previous one, allowing the maneuver to take place at a time later than at t=0. The new model also features a control settings area the puts all the control system options in one dedicated place. Other than that nothing else was changed.

STARK INDUSTRIES

2. LIDAR Data Variation



Mean	0.908 m
Variance	1.84E-5 m^2
Standard Deviation	0.00429 m

The variation of the measured LIDAR data is shown on the histogram to the left. The standard deviation is close to 4 centimeters. As I am not completely familiar with LIDAR and its accuracy I cannot say whether that is good or not, but it equates to about 5% error with the mean at the 90 centimeter mark.

Looking closer at the data though, there are two peak in the histogram data. This is interesting as these is about an 8 millimeter difference between the two peaks, while the average is a valley. This looks closer to a bimodal distribution rather than unimodal gaussian distribution, but more statistical test would have to be calculated to know for sure.

3. Kalman Filter Development

As the data is fairly noisy, the data must be filtered to get a more accurate estimate of the altitude of the bicopter from the lidar data. The equations used are shown to the right.

This model assumes several things:

- The user knows that the bicopter's zero is around 90 centimeters and will hover around that point when demanded to hover at 0 meters.
- The bicopter is hovering nearly still.

With these assumptions the constants and initial conditions to the right can be assumed.

$$Gain = \frac{E_{est}}{E_{est} + E_{sensor}}$$
$$EST_t = EST_{t-1} + Gain\,(SEN - EST_{t-1})$$
$$E_{est_t} = (1 - Gain)\left(E_{est_{t-1}} + E_{process}\right)$$

Where:

E_{EST} = Error in the estimate

E_{sensor} = Sensor error

EST_t = Estimate at point in time

$Gain$ = Kalman gain

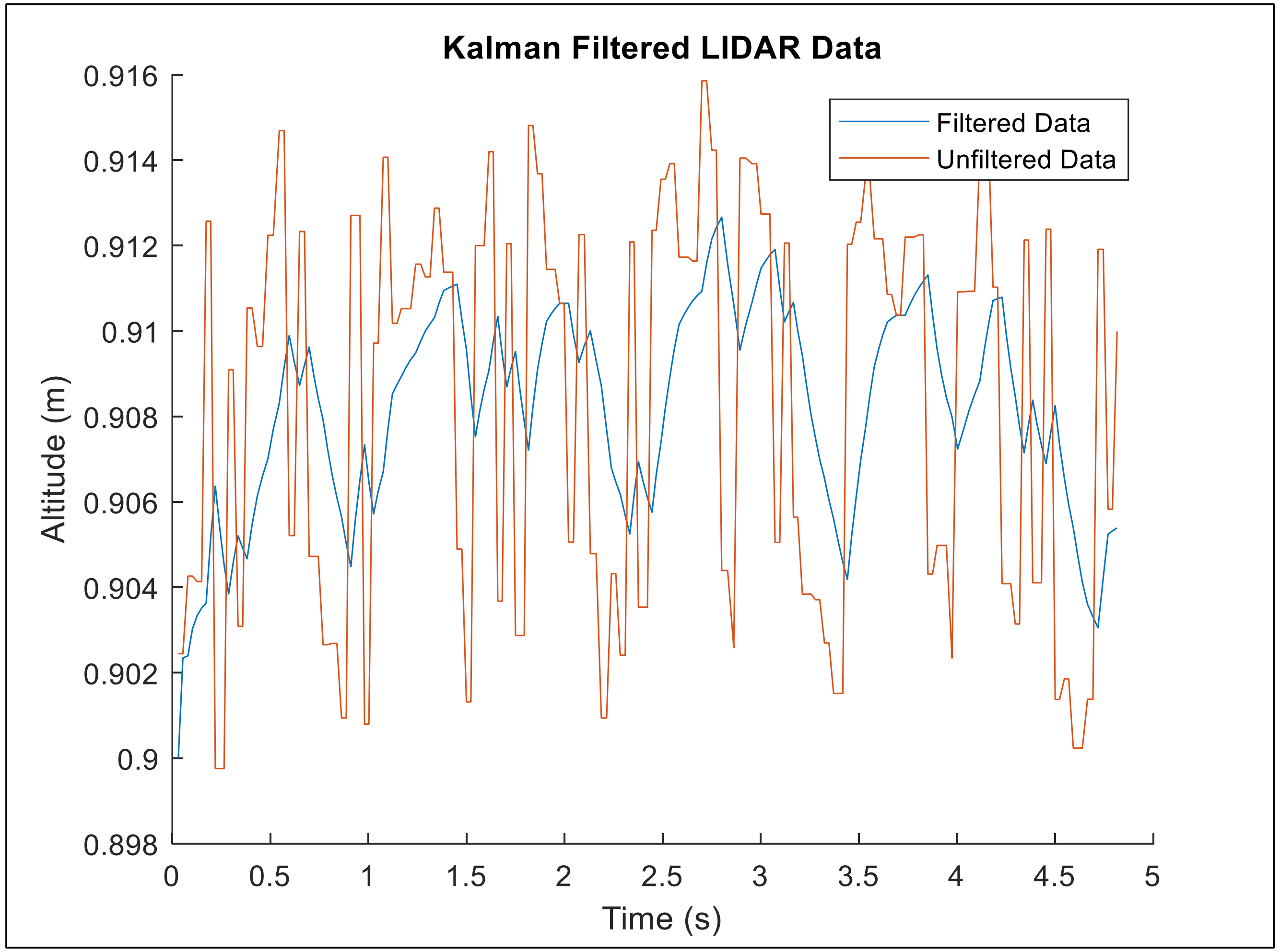
Sen = Sensor measurement

E_{sensor} = 0.0043 meters

$E_{process}$= 0.0001 meters

$$EST_0 = 0.9\,meters$$
$$E_{est_0} = 0.1\,meters$$

4. Kalman Filtered LIDAR Data

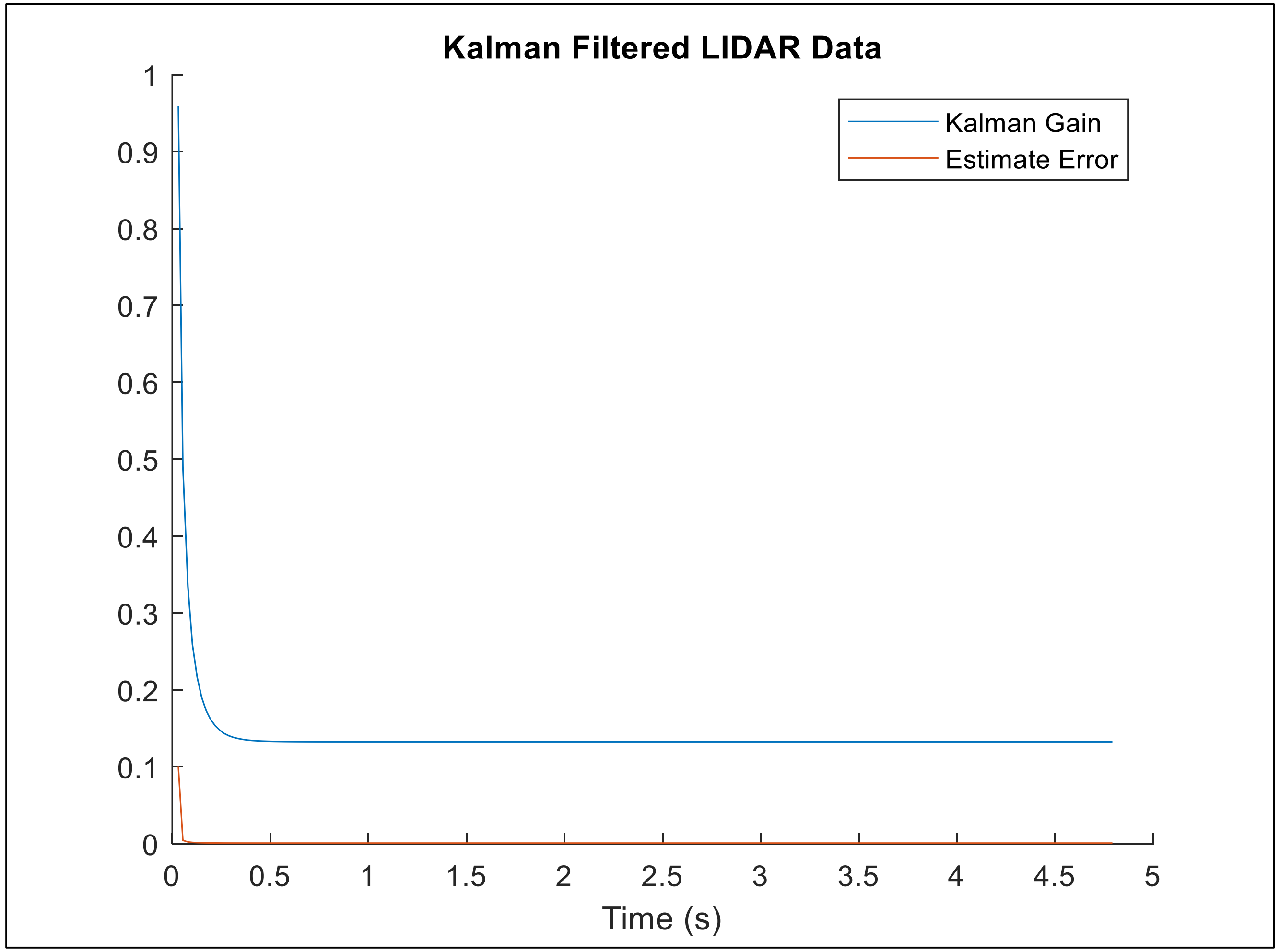


The left shows the LIDAR data after being filtered by a one dimensional Kalman filter. The initial condition of 0.9 meters can be seen on the left side of the graph, and since the process data has more weight than the sensor data, the estimate slowly rises to near the middle of the sensor data. But since the sensor data also has weight the estimate changes with it. This change is within one centimeter, which I think is a reasonable estimate for a bicopter hovering.

If the bicopter was known to be hovering perfectly still, the process model would have an error of zero and the filtered data would asymptotically reach a value with very little variation, but this is not a good assumption in real life

STARK INDUSTRIES

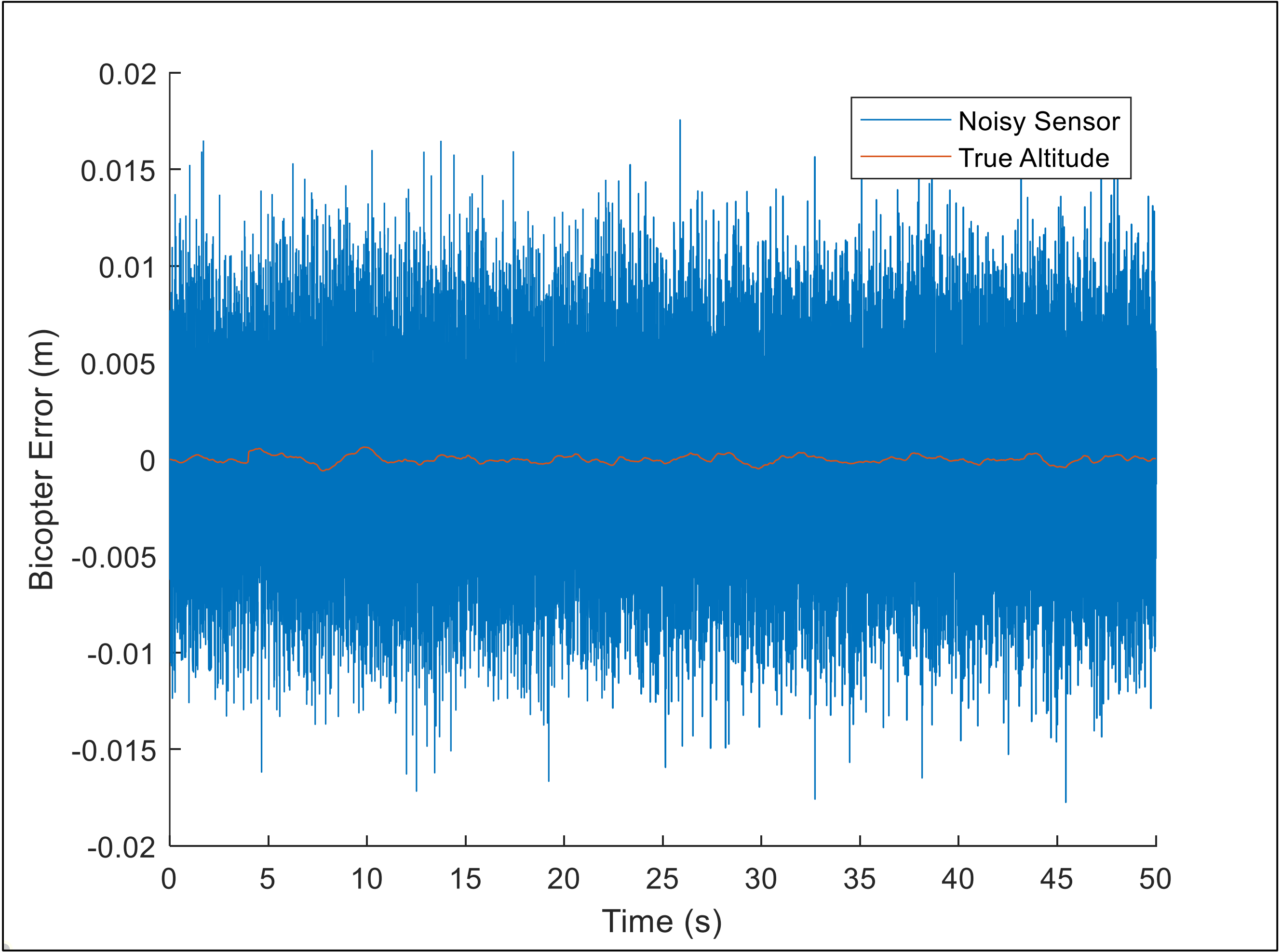
5. Kalman Filtered LIDAR Data Gains



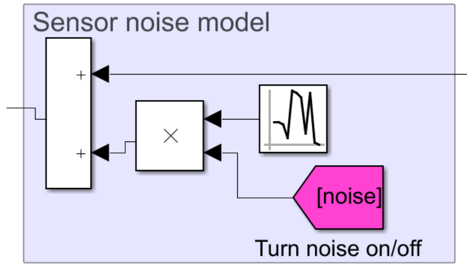
The left plot shows the Kalman gain and the estimation error of the previous slide's filtered data. It shows that the Kalman gain is at first near unity, which means that the sensor's measurement is weighted more heavily than the process's estimation. This makes sense as the initial condition is just a guess, so the filter relies more on the sensor's data at first rather than the initial guess by the user.

The estimate error starts at the initial error value given, then quckly decreases to near zero, meaning the fitler calculates confidence in the estimation it is producing given the process and sensor errors given.

6. Effect of Noise on PID and FF Control

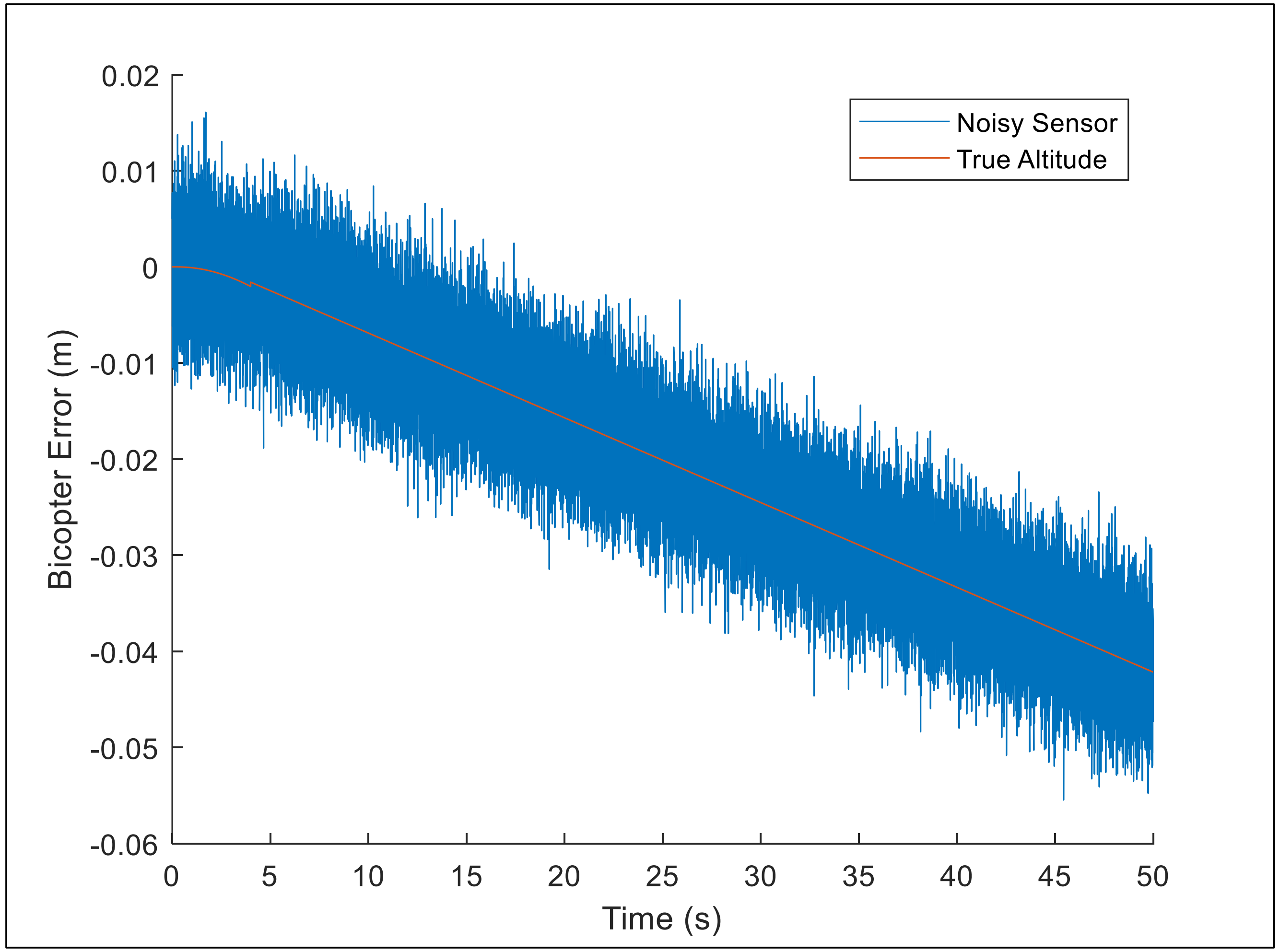


The Simulink model was edited to include noise to replicate a sensor with noise as measured with the LIDAR sensor previously. This was implemented with the following block (with the LIDAR sensor's variance:



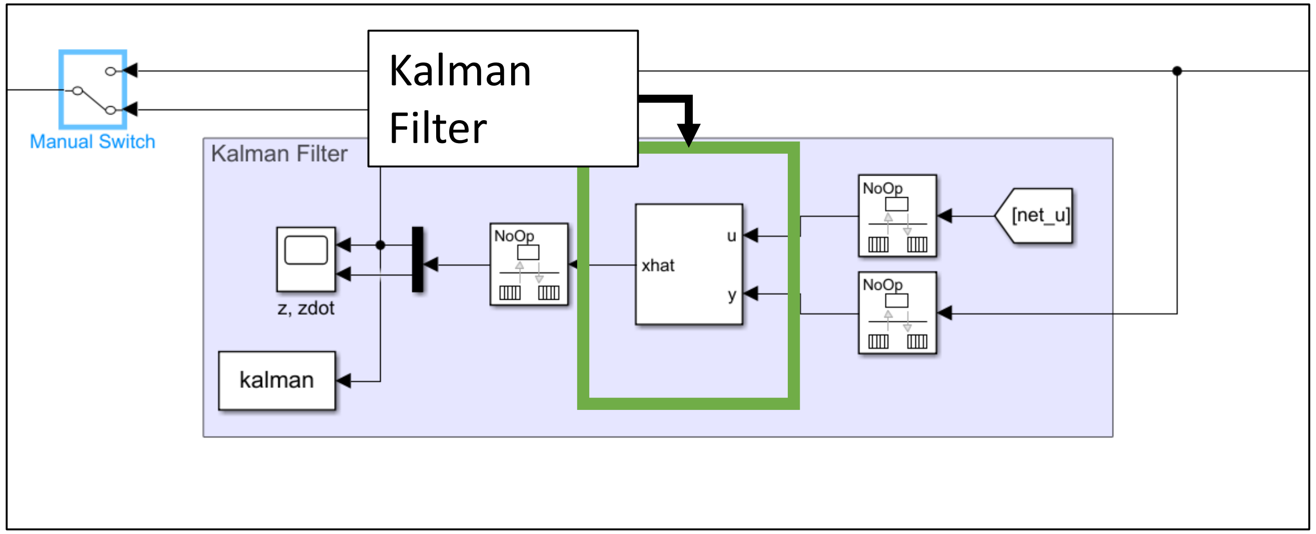
With the noisy sensor the true altitude varies by remarkably little. This is primarily due to the integrated filter on the derivative term of the PID controller, which is a low-pass filter that is able to filter out much of the noise. The average error from the demanded altitude is much less than one millimeter.

7. Effect of Noise on Feedforward Control



Because the feedforward controller doesn’t rely on the sensor input, it was not expected to change due to the noisy sensor input. This was proven true, as the true altitude is still imperfect from the precomputed impulse not correcting for the minute errors. The true altitude drifts with time as well as the sensor’s output.

8. Kalman Filter Implementation

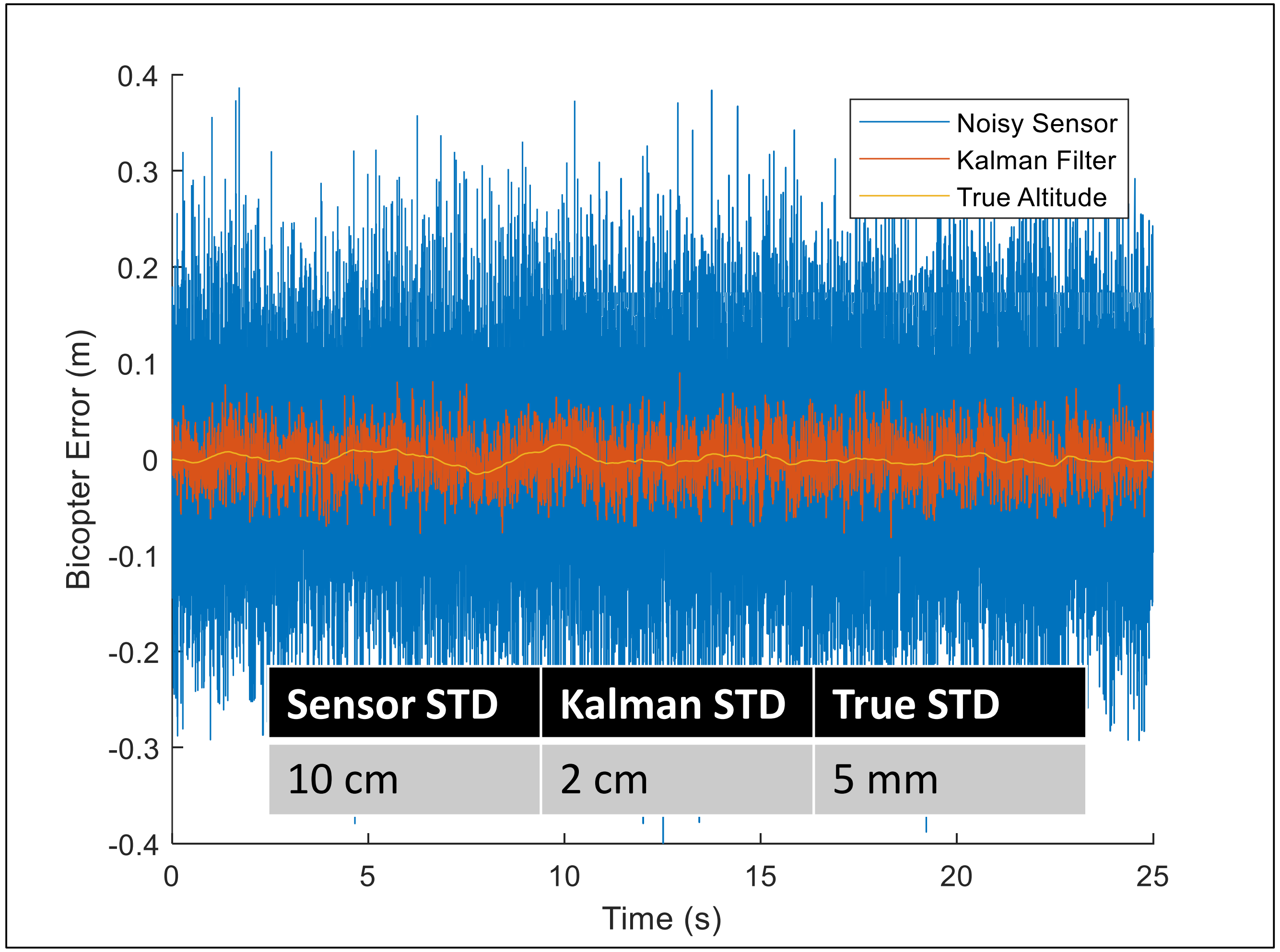


To obtain the best and most accurate data from the noisy sensor for the bicopter's feedback control system, a Kalman filter was implemented in the Simulink model. The primary component is the Kalman filter block in the center. The parameters used are to the right. The process and sensor covariance change with each scenario shown. The sensor covariance for each scenario is 0.01 instead of the much lower covariance of the LIDAR sensor to exemplify the Kalman filter

A	B	C	D	X_0	N
$\begin{bmatrix} 1 & 0.001 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{0.001^2}{2 \times 1.23} \\ \frac{0.001}{1.23} \end{bmatrix}$	$[1 \quad 0]$	0	$[0 \quad 0]$	0

STARK INDUSTRIES

9. Kalman Filtered Sensor Input 1



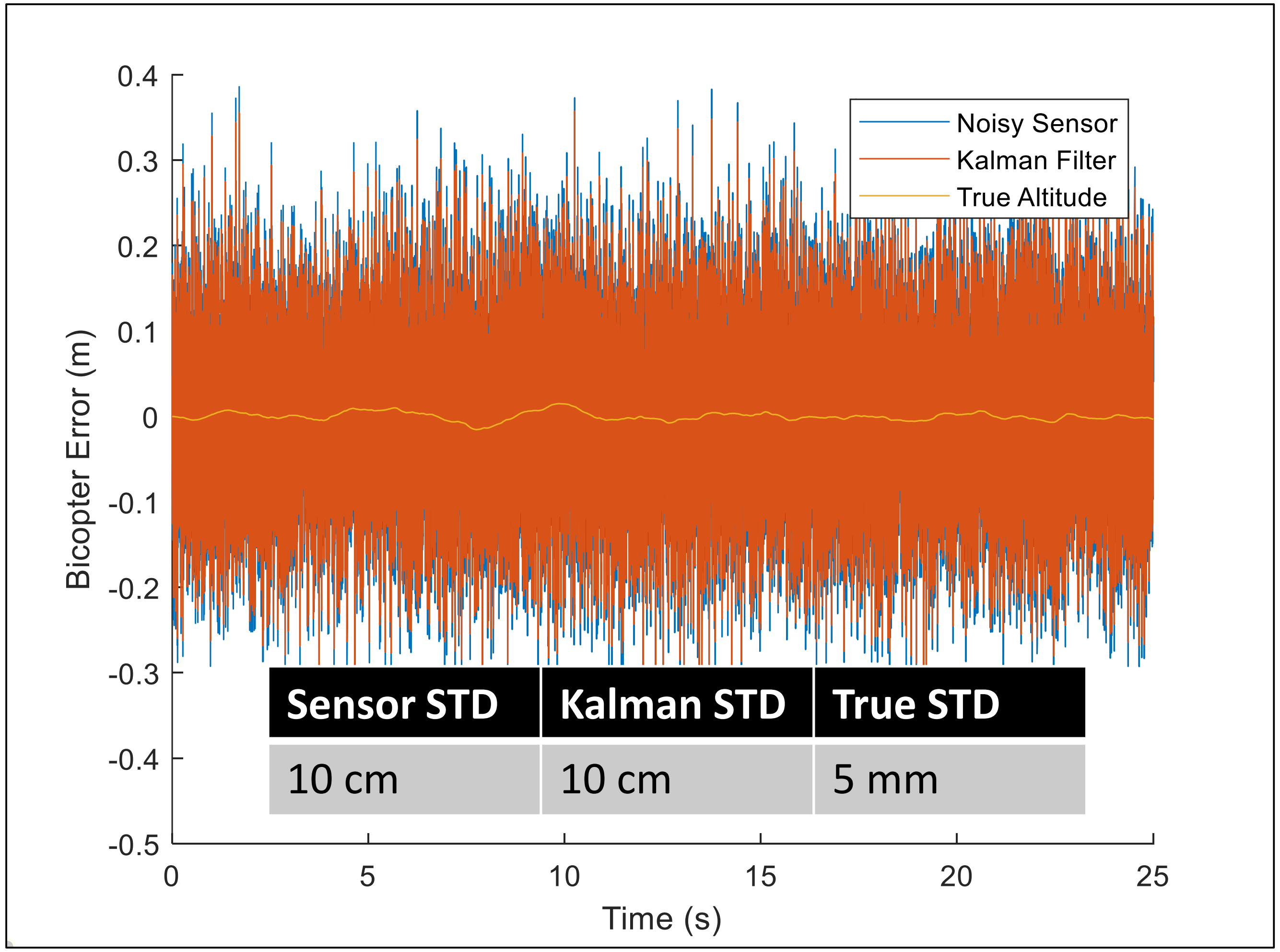
Sensor STD	Kalman STD	True STD
10 cm	2 cm	5 mm

For this scenario the Kalman filter that was implemented had a process covariance of 0.0001, indicating confidence in the process model, and a sensor covariance of 0.01, which is less accurate than the LIDAR sensor but still relatively accurate in the grand scheme.

The measured sensor data is by far the most noisy, with a standard deviation from the demanded altitude of just about 10 centimeters. The Kalman filter is the next most noisy, but much less than the sensor with a standard deviation of error around 2 centimeters. The PID control's derivative filter seems to have been able to filter out most of the Kalman filter's noise so the true altitude is the least noisy with the standard deviation of error around 5 millimeters. This is very low and accurate, even compared to the previous unfiltered sensor data as the covariance of the sensor has been increased several orders of magnitude, but the true error didn't scale as much.

STARK INDUSTRIES

10. Kalman Filtered Sensor Input 2

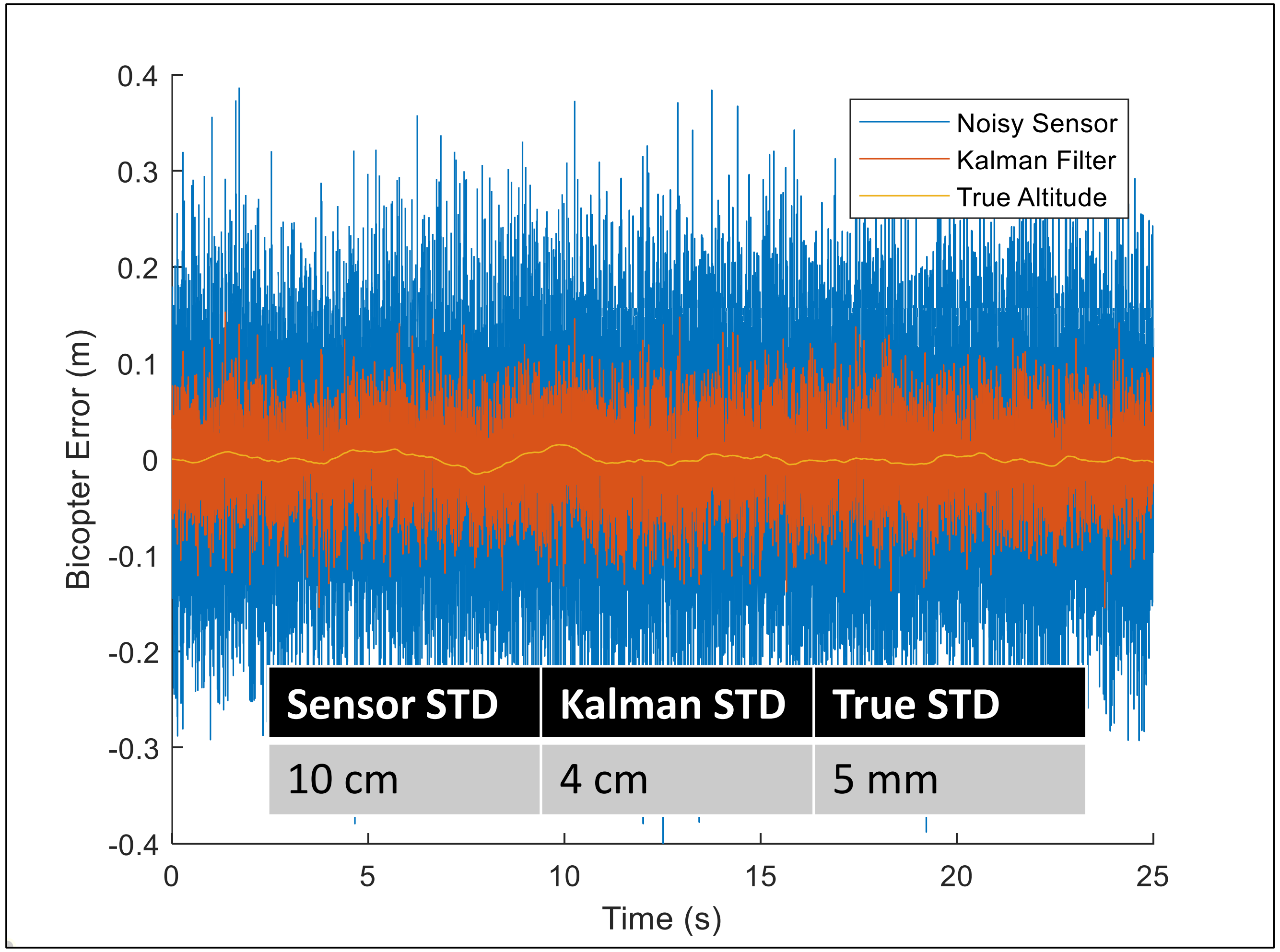


Sensor STD	Kalman STD	True STD
10 cm	10 cm	5 mm

For this scenario the Kalman filter that was implemented had a process covariance of 0.1 and a sensor covariance of 0.01. This means that the sensor has much more weight in this scenario compared to the previous, meaning that the true altitude will be more effected by the relatively large sensor error.

This is confirmed by the plot to the left, as the Kalman filtered data is almost exactly the same as the filtered data. The savior for this scenario is the PID controller's derivative filter that filters out much of the noise. If the derivative filter is removed the system very quickly becomes unstable with large oscillations, much quicker than the previous scenario. The standard deviation from the demanded altitude is around 5 millimeters which is still fairly low, but visually it is moving around more.

11. Kalman Filtered Sensor Input 3



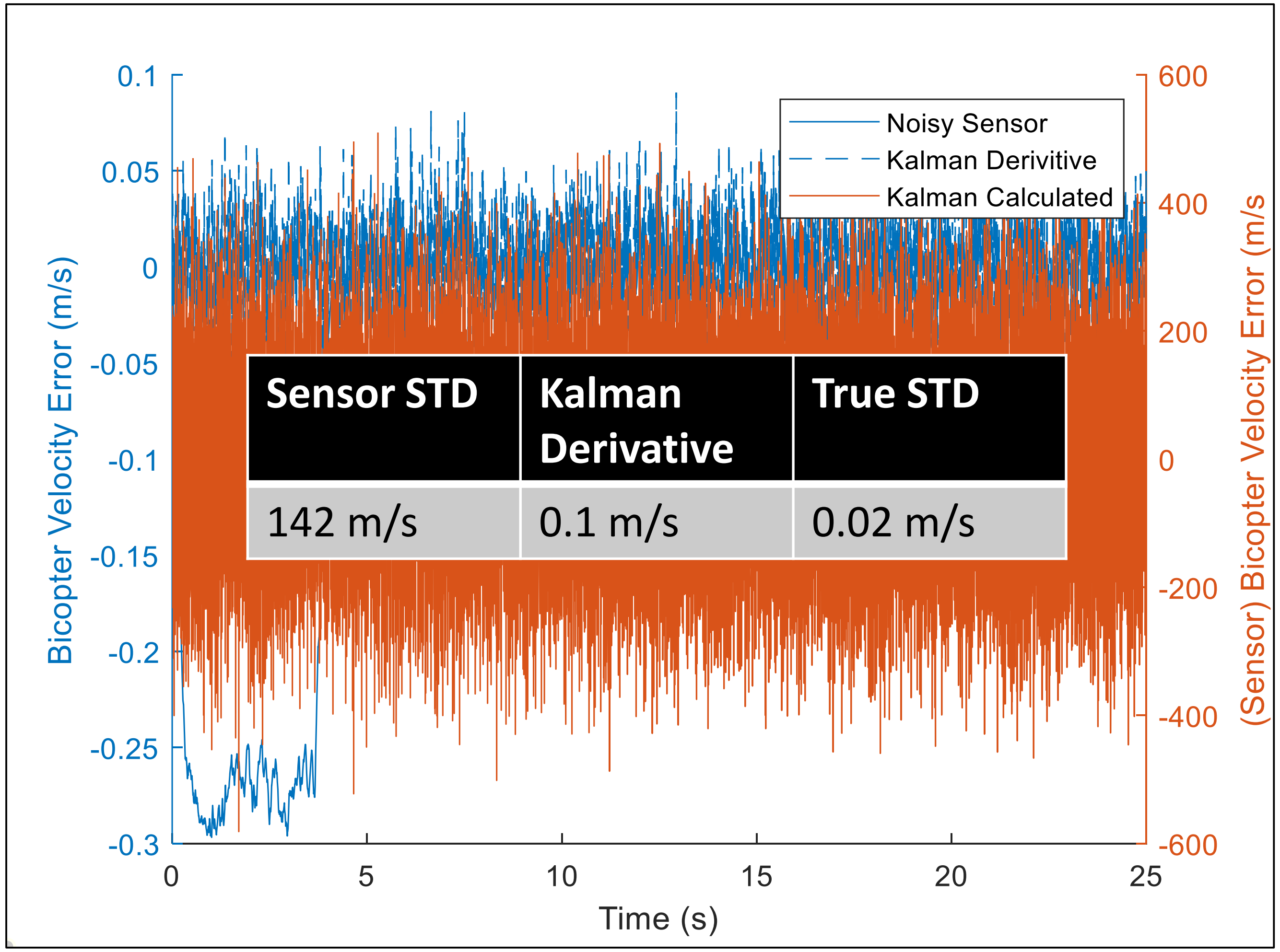
Sensor STD	Kalman STD	True STD
10 cm	4 cm	5 mm

For this scenario the Kalman filter that was implemented had a process covariance of 0.1 and a sensor covariance of 1.0, even though it is known that the covariance is 0.01. Even though the process covariance hasn't changed since the previous scenario this will lead to the sensor data having less weight as the sensor covariance is relatively larger compared to the process covariance in this scenario.

The plot to the left confirms this hypothesis, and the Kalman filtered data is much lower in variation compared to the previous, around 4 centimeters compared to the 10 centimeters previously. This means that the Kalman filtered altitude is not very dependent on the magnitude of the covariances, but their relative values to each other.

STARK INDUSTRIES

11. Velocity Measurement Exploration



Sensor STD	Kalman Derivative	True STD
142 m/s	0.1 m/s	0.02 m/s

The bicopter does not have a velocity sensor directly on it, but the velocity is useful to know for either telemetry or control. Three options in determining the velocity of the copter were explored: taking the derivative of the noisy sensor data, taking the derivative of the Kalman filtered altitude, and directly using the Kalman filtered calculated velocity. The plot to the left shows the error from the bicopter's true velocity. The left y-axis is the two Kalman filter-based option the right is the sensor-based velocity.

The noisy sensor is slightly visible on the back, but the main tell of the noisy sensor is that the y-axis on the left is the sensor derivative velocity, as if it was on the same axis the other velocities would not be able to be seen. It's average error is near 150 m/s, which is completely unacceptable. The Kalman derivative data is almost fine but at the beginning, the blue rectangle line going down is that signal. IT's average error is around 0.1 m/s. The velocity calculated by the Kalman filter directly is the most accurate, with an average error of around 2 cm/s, which is very low for not being measure directly. Because of this, the Kalman filter calculated velocity is the best option.