Deriving Kalman Filter via Orthogonal Projection Lemma

Introduction

This Jupyter Notebook presents the derivation of the Kalman Filter using the Orthogonal Projection Lemma. The Kalman Filter is a recursive algorithm for state estimation in dynamic systems, and understanding its derivation is crucial for grasping its underlying principles.

Orthogonal Projection Lemma

The Orthogonal Projection Lemma is a key concept in the derivation of the Kalman Filter. It involves projecting a vector onto a subspace in a way that minimizes the orthogonal distance. The Kalman Filter leverages this lemma to update state estimates based on measurements.

```python

Implementation of the Orthogonal Projection Lemma for Kalman Filter

...

Kalman Filter derivation steps

...

In [1]:

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt

# Generate synthetic data
np.random.seed(42)
time = np.arange(0, 10, 0.1)
true_state = 2 * time + 1
measurement_noise = np.random.normal(0, 2, len(time))
measurements = true_state + measurement_noise

# Kalman Filter Derivation using Orthogonal Projection Lemma
def kalman_filter_derivation(measurements, initial_state_estimate, initial_error_covarian
ce, process_noise_covariance, measurement_noise_covariance):
    state_estimate = [initial_state_estimate]
    error_covariance = [initial_error_covariance]

    for measurement in measurements:
        # Prediction Step
        predicted_state_estimate = state_estimate[-1]
        predicted_error_covariance = error_covariance[-1] + process_noise_covariance

        # Update Step
        kalman_gain = predicted_error_covariance / (predicted_error_covariance + measure
ment_noise_covariance)
        updated_state_estimate = predicted_state_estimate + kalman_gain * (measurement -
predicted_state_estimate)
        updated_error_covariance = (1 - kalman_gain) * predicted_error_covariance

        # Store results
        state_estimate.append(updated_state_estimate)
        error_covariance.append(updated_error_covariance)
```

```
    return state_estimate, error_covariance

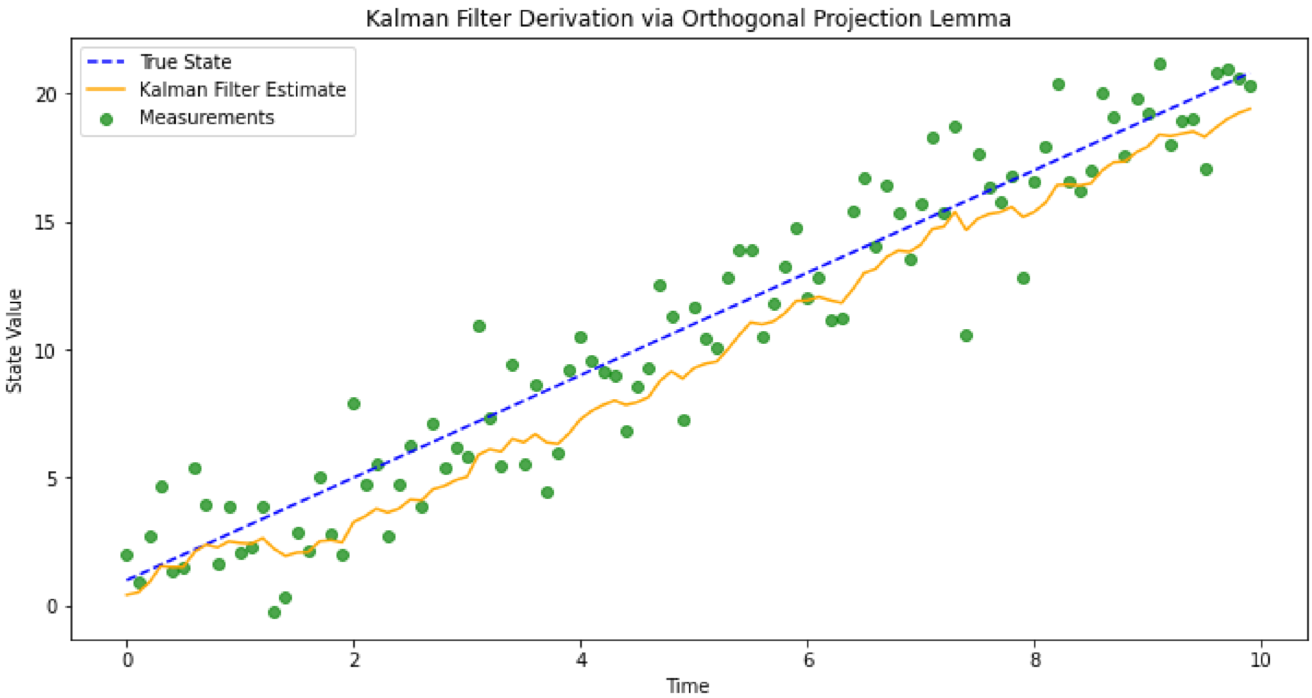
# Initial parameters
initial_state_estimate = 0
initial_error_covariance = 1
process_noise_covariance = 0.1
measurement_noise_covariance = 4

# Apply Kalman Filter Derivation
state_estimate, error_covariance = kalman_filter_derivation(measurements, initial_state_e
stimate, initial_error_covariance, process_noise_covariance, measurement_noise_covariance
)

# Plotting
plt.figure(figsize=(12, 6))

plt.plot(time, true_state, label='True State', linestyle='--', color='blue')
plt.scatter(time, measurements, label='Measurements', color='green', alpha=0.7)
plt.plot(time, state_estimate[1:], label='Kalman Filter Estimate', color='orange')

plt.title('Kalman Filter Derivation via Orthogonal Projection Lemma')
plt.xlabel('Time')
plt.ylabel('State Value')
plt.legend()
plt.show()
```



Conclusion

In conclusion, this notebook has walked through the derivation of the Kalman Filter using the powerful Orthogonal Projection Lemma. Key concepts such as the prediction step, update step, Kalman gain, and state estimation have been explored in detail. The Orthogonal Projection Lemma plays a fundamental role in the Kalman Filter, allowing it to optimally update state estimates based on noisy measurements.

Key Takeaways:

- **The Kalman Filter is a recursive algorithm for state estimation in dynamic systems.**
- **The Orthogonal Projection Lemma is leveraged to project vectors onto subspaces, minimizing orthogonal distances.**
- **Prediction and update steps in the Kalman Filter allow for optimal state estimation in the presence of noise.**
- **The Kalman Filter is widely used in various fields, including control systems, navigation, and signal processing.**

Next Steps:

Next Steps:

- Explore more advanced Kalman Filter variants, such as the Extended Kalman Filter for non-linear systems.
- Implement the Kalman Filter in real-world applications and observe its performance.
- Contribute to open-source projects related to Kalman Filter implementations and applications.

This notebook serves as a starting point for understanding the derivation of the Kalman Filter. Further exploration and hands-on experience will deepen your appreciation of this powerful algorithm in the realm of state estimation.