Code explanation:

1. **Import and create the model**: You import the COMSOL classes and create the 3D model where the simulation will be developed.
2. **Geometry**: They define the environment in which the laser will propagate. In this case, there is a block that simulates the object that the LIDAR must detect.
3. **laser source**: A laser is created with a wavelength of 905 nm, which is common in LIDAR systems. The number of rays and power are also specified.
4. **Environmental conditions**: The propagation of the laser in air is simulated.
5. **Detector**: The detector is placed in the same position as the object to record the laser reflection.
6. **Transient simulation**: It is specified that the simulation is a function of time (transient), which is important to measure the time of flight of the laser.
7. **Results**: A display of the reflected rays is generated and distances are calculated based on flight time.

Effectiveness Adjustments:

To improve the effectiveness of the LIDAR system, you can adjust several parameters in the code:

- **Scan angle**: You can add a loop that changes the laser emission angle, simulating a sweep in 3D space.
- **Noise and precision**: To simulate real conditions, you can add a noise component to the distance data to evaluate how the system responds to disturbances.
- **Pulse rate**: Increase the repetition rate to obtain more data per second and improve resolution.

COMSOL-MATLAB connection:

- **Advanced LIDAR**: If you want to integrate real-time data processing, you can extend the code in MATLAB to process the data obtained in COMSOL, such as doing signal filtering, identifying objects, etc.
- **Simulation of dynamic scenarios**: You can simulate moving objects and calculate relative speed using the change in flight time.

Effectiveness Variables:

1. **System resolution**: Depends on the pulse frequency and the number of rays emitted.
2. **Maximum range**: It is determined by the power of the laser and the ability of the detector to collect signals over long distances.

3. **Distance Accuracy**: Noise in the environment can affect flight time, so you can add noise simulations to see how accuracy varies.

This is a basic layout that you can expand. If you have more details you would like to include, such as material types or different geometries

- **Specs**:
  - **High dynamic range** to be able to measure signals reflected from different surfaces with different reflectivities.
  - **Sensitivity**: Sensitivity should be adjusted to detect reflections from low reflectivity surfaces such as water or asphalt.

c) Optoelectronics (Beam Splitter and Lenses)

- **Function**: Manages the propagation of the light beam and the reflection towards the photodetector. In aircraft LIDAR systems, optical components such as the beam splitter are crucial to direct the beam correctly and ensure that the reflected rays reach the detector without distortion.
- **Specs**:
  - **Beam Splitter**: It should be made of high transparency material in the near infrared (silicon dioxide or germanium).
  - **Glasses**: Lenses **silicon** the **sapphire** which can withstand high temperatures and reduce optical aberrations.

2. Important materials for a LIDAR system on an airplane

- **Silicon Dioxide (SiO2)**: Used for lenses and beam splitters due to its low absorption and high transparency in the infrared range.
- **Germany (Ge)**: Suitable for lenses and detectors in LIDAR systems, as it is an excellent conductor of infrared signals.
- **Silicon and Germanium Photodiodes**: They are key for the photodetector, given their high quantum efficiency in the near-infrared wavelength range.
- **Aluminum or titanium alloys**: For the chassis and other structural components of the LIDAR system, since they are lightweight and strong, which is crucial to minimize overall weight in aeronautical applications.

3. Wave geometries and fitting a realistic model

The **wave geometries** and the way the beam propagates in an aircraft's LIDAR system are crucial to the accuracy of the measurements.

a) Laser wave propagation

- **Beam propagation simulation**: In **ANSYS Lumerical** the **MATLAB**, you can model the propagation of a Gaussian or collimated beam over long distances to ensure that the beam maintains its coherence over long distances.
- **Beam divergence**: Beam divergence should be small enough to avoid loss of resolution, but large enough to cover an adequate scan area.

b) Simulation in ANSYS or MATLAB

Below I show you a code in **MATLAB** which could be used to simulate laser beam propagation in a LIDAR system in an aeronautical environment, with special emphasis on the parameters of **effectiveness** of the laser, such as wavelength, power and resolution.

- **Specs:**
  - **High dynamic range** to be able to measure signals reflected from different surfaces with different reflectivities.
  - **Sensitivity:** Sensitivity should be adjusted to detect reflections from low reflectivity surfaces such as water or asphalt.

c) Optoelectronics (Beam Splitter and Lenses)

- **Function:** Manages the propagation of the light beam and the reflection towards the photodetector. In aircraft LIDAR systems, optical components such as the beam splitter are crucial to direct the beam correctly and ensure that the reflected rays reach the detector without distortion.
- **Specs:**
  - **Beam Splitter:** It should be made of high transparency material in the near infrared (silicon dioxide or germanium).
  - **Glasses:** Lenses **silicon** the **sapphire** which can withstand high temperatures and reduce optical aberrations.

2. Important materials for a LIDAR system on an airplane

- **Silicon Dioxide (SiO2):** Used for lenses and beam splitters due to its low absorption and high transparency in the infrared range.
- **Germany (Ge):** Suitable for lenses and detectors in LIDAR systems, as

it is an excellent conductor of infrared signals.

- **Silicon and Germanium Photodiodes:** They are key for the photodetector, given their high quantum efficiency in the near-infrared wavelength range.
- **Aluminum or titanium alloys:** For the chassis and other structural components of the LIDAR system, as they are lightweight and strong, which is crucial to minimize overall weight in aeronautical applications.

3. Wave geometries and fitting a realistic model

The **wave geometries** and the way the beam propagates in an aircraft's LIDAR system are crucial to the accuracy of the measurements.

a) Laser wave propagation

- **Beam propagation simulation:** In **ANSYS Lumerical** the **MATLAB**, you can model the propagation of a Gaussian or collimated beam over long distances to ensure that the beam maintains its coherence over long distances.
- **Beam divergence:** Beam divergence should be small enough to avoid loss of resolution, but large enough to cover an adequate scan area.

b) Simulation in ANSYS or MATLAB

Below I show you a code in **MATLAB** which could be used to simulate laser beam propagation in a LIDAR system in an aeronautical environment, with special emphasis on the parameters of **effectiveness** of the laser, such as wavelength, power and resolution.

Code in Matlab:

```
% LIDAR system parameters
lambda = 1550e-9;  % Laser wavelength in meters (1550 nm)
power = 10;        % Laser power in watts
divergence = 0.001; % Beam divergence in radians
distance = 5000;   % Maximum distance in meters (5 km)
speed_of_light = 3e8;  % Speed of light in m/s

% Calculate the radius of the laser beam at the maximum distance
beam_radius = divergence * distance;

% Calculate beam area at maximum distance
beam_area = pi * (beam_radius)^2;

% Laser Time of Flight (ToF)
time_of_flight = 2 * distance / speed_of_light;  % ToF en segundos
```

```
% Spatial resolution calculation
resolution = lambda / (2 * beam_radius);  % System Resolution

% Show key results
fprintf('Radio del haz a 5 km: %.3f m\n', beam_radius);
fprintf('Área del haz a 5 km: %.3f m^2\n', beam_area);
fprintf('Time of flight (ToF): %.3f µs\n', time_of_flight * 1e6);
fprintf('Spatial resolution: %.3f m\n', resolution);
```

Explanation:

1. **beam radius**: Calculates how the laser beam expands as it travels up to 5 km.
2. **Beam area**: Shows the area covered by the beam at that distance.
3. **Time of Flight (ToF)**: Calculates the time it takes for the laser to travel to an object 5 km away and return to the sensor.
4. **Spatial resolution**: Defines the precision of the system based on the wavelength and beam size.

4. Adjustments for a realistic and practical model

To fit a LIDAR model to a realistic production line in an airplane like those in **Airbus**, the following settings are critical:

a) Redundancy and Security

- **Redundant systems**: To avoid failures in mid-flight, multiple light sources and detectors are implemented. The simulation must take into account multiple ray paths to ensure that the system continues to function if one of the lasers fails.
- **noise filtering**: The system should be designed to remove ambient noise from the data. This can be simulated in **MATLAB** with post-processing signal filtering.

b) Plane Geometry

- **Placing the LIDAR system on the aircraft**: The design must consider integration with the aerodynamic geometry of the aircraft fuselage, ensuring that it does not interfere with other functions of the aircraft, and minimizing air resistance.

c) Realistic Environment Simulation

- **atmospheric factors**: In **ANSYS** the **MATLAB**, you must simulate weather effects, such as rain, fog, and turbulence, which can scatter the beam and affect measurements. For this, variables such as absorption and dispersion can be added to the simulation.

This design can be expanded by adding more details and configurations to achieve a more complete and realistic model.

Redundancy Systems and Implementation of Multiple Light Sources in Airborne LIDAR

The **redundancy** It is crucial in airborne lidar systems to ensure that measurements continue to work even if a part of the system fails. This involves using multiple light sources and photodetectors, ensuring that the system can operate continuously during flight and provide reliable data, even in difficult conditions.

1. Redundancy with Multiple Light Sources

In an environment such as an airplane **Airbus**, various laser light sources allow:

- **Complete field of view coverage**: Multiple sources can scan larger or specific areas of the aircraft environment.
- **Measurement Overlay**: This helps obtain multiple readings from the same point to average measurements and reduce errors, which improves accuracy. **precision**.
- **Failure resilience**: If one laser or photodetector fails, the others can continue operating and covering critical areas.

Technical Implementation:

- **Redundant light sources**: Two or more lasers with different orientations can be used to cover the entire field of view of the aircraft. An example would be having one forward and one downward for terrain mapping.
- **Split laser beam**: The use of a **beam splitter** Splitting a laser beam into multiple paths can also serve as a form of redundancy.

2. Electronic Devices Valid for Aircraft

Devices used on an aircraft must meet strict safety requirements. **weight, energy consumption, resistance to vibrations, temperature fluctuations** and ability to operate at high altitudes. Here are some key components:

a) Laser Light Sources

- **High power diode lasers (905 nm or 1550 nm)**: They are ideal due to their low weight and compact size. Additionally, 1550 nm is safe for the eyes, which is important in LIDAR applications.
- **Fiber optic lasers**: They can handle higher powers and wavelengths, ideal for long-range and high-altitude applications, as they are less susceptible to atmospheric dispersion.

b) Detectors

- **Avalanche Photodiodes (APD)**: They are widely used in LIDAR systems due to their high sensitivity and ability to operate over long distances.
- **InGaAs-based Semiconductor Detectors**: They are optimal for the detection of wavelengths in the range of 900-1700 nm, essential in an environment with multiple light sources.

c) Processors

- **Field-Programmable Gate Arrays (FPGAs)**: The **FPGAs** They are used to process large volumes of data in real time, especially in the integration of multiple signals from different laser sources and photodetectors.
- **Low power microcontrollers**: Microcontrollers like those in the series **ARM Cortex** They offer a balance between power and low consumption, essential for systems that operate on airplanes.

d) Inertial Sensors and GPS

- These devices complement the LIDAR system measurements, helping to correct data based on the aircraft's movements.

3. Precision in Noise Filtering

He **noise** in a LIDAR system it can come from several sources, including:

- **Environmental noise**: Sunlight, other electronic systems.
- **Electromagnetic interference (EMI)**: Aircraft systems generate interference that can affect lidar readings.
- **Thermal noise**: At high altitudes, temperature fluctuations can cause noise in the detectors.

a) Filtering Techniques

A good LIDAR system should implement advanced techniques to filter out noise and improve accuracy. Some techniques include:

Digital Filters:

1. **Kalman filter**: Ideal for dynamic systems such as an airplane, where positions and speeds are constantly changing. The Kalman filter adjusts measurements based on the estimated state of the system, reducing noise without losing relevant information.
   - **Advantages**: Improves accuracy in situations where there is noisy or uncertain data. It can also predict the next state of the system.

```
% Kalman Filter Example in MATLAB
% Simulated noisy measurement (position of an object)
measured_position = [1.2, 2.1, 3.5, 4.6, 5.1] + randn(1,5) * 0.2;

% Kalman filter parameters
estimated_position = 0;  % Estimated starting position
velocity = 1;  % Constant speed of the object
P = 1;  % Initial estimated error
R = 0.01;  % Measurement noise
Q = 0.001;  % Model noise

for t = 1:length(measured_position)
    % Prediction
    estimated_position = estimated_position + velocity;
    P = P + Q;

    % Update
    K = P / (P + R);  % Kalman Gain
    estimated_position = estimated_position + K * (measured_position(t) -
estimated_position);
    P = (1 - K) * P;

    % Show estimate
    fprintf('Estimated position: %.3f\n', estimated_position);
end
```

2. **Median and Weighted Average Filters**: Used to filter out extreme noise peaks. In an airplane, LIDAR measurements may have reflections from metallic surfaces or other shiny objects. The median filter removes these outliers.

3. **LIDAR Signal Processing**: Real-time processing using FPGAs can apply techniques such as **wavelet filtering** to reduce high frequency noise while preserving important signal details.

b) Interference Suppression Techniques

- **Optical filtering**: Bandpass filters are used in the receiver to suppress unwanted wavelengths (sunlight, for example).
- **Pulse Modulation**: By modulating the laser pulse, it is possible to distinguish LIDAR signals from other light sources, reducing interference.

c) Vibration Compensation

- In an aircraft, the LIDAR system must compensate for mechanical vibrations. This is achieved using **inertial sensors** combined with the LIDAR system to adjust measurements in real time and ensure accuracy despite the constant movement of the aircraft.

4. Design of a Realistic Model for Production

When implementing a LIDAR system on an Airbus aircraft, the system must be modular and scalable. Here are some key points for designing a realistic model:

a) Modularity

- **Replaceable modules**: Each light source and detector should be individually replaceable for maintenance or upgrade, without needing to replace the entire system.

b) Weight and Energy Optimization

- Since the system will be mounted on an airplane, the **weight** and the **power consumption** They should be minimal. Use low-power diode laser light sources and efficient processors (such as **FPGAs**) is key.

c) Integration with Aircraft Systems

- LIDAR must be integrated with the aircraft's navigation systems, GPS, and inertial sensors to obtain more accurate measurements and correct for error based on the aircraft's altitude and position.

Summary:

- **Redundancy**: It can be implemented with multiple light sources and detectors, using a beam splitter to split the beam or with laser sources oriented in different directions.

- **electronic devices**: Use diode lasers, APD photodiodes, FPGA processors, and electronic components optimized for an airborne environment.
- **noise filtering**: Techniques such as Kalman filter, median filters, pulse modulation and optical filtering ensure accuracy and reduce interference.

This ensures a system **Rugged LIDAR** and accurate for operation in complex aeronautical environments such as a **Airbus**. Would you like to continue with more technical details or a diagram of the LIDAR system?

In MATLAB, to implement an airborne LIDAR system, there are several modules and functions that you can use. Here is a detailed approach on how you could design and simulate a LIDAR system, with redundancy techniques, noise filtering, and signal analysis.

1. Relevant Modules in MATLAB

To work with LIDAR in MATLAB, you will need to use the following tools and libraries:

a) Signal Processing Toolbox

It allows lidar signal filtering, noise reduction and data processing to detect peaks in reflected signals.

b) Communications Toolbox

Used to work with modulated signals, time synchronization and signal processing from multiple sources.

c) MATLAB LIDAR Toolbox

If you have access to the **LIDAR Toolbox**, you can take advantage of specialized functions to process and analyze LIDAR-generated 3D point cloud data, such as object segmentation and data fusion.

d) Control System Toolbox

It helps you implement filters like **Kalman filter** to improve the precision of the system against noise and measurement errors.

Design of a LIDAR System in MATLAB

Next, I detail how you could simulate and analyze a LIDAR system using different modules in MATLAB. The focus is on creating a system with multiple light sources, redundancy techniques and noise filtering, specifically oriented for applications in an airborne environment such as an aircraft.

1. Modules Used

- **Signal Processing Toolbox**: For the generation and analysis of LIDAR signals.
- **Control System Toolbox**: To apply filtering techniques such as **Kalman filter** for noise reduction.
- **Lidar Toolbox**: For processing and displaying 3D point cloud data (if available).

2. Components of the LIDAR System in MATLAB

a) LIDAR Signal Generation

To simulate a LIDAR system, a laser pulse signal is first generated. The reflected signal (back to the sensor) is what is analyzed to measure the distance to the object.

matlab

```
% Laser parameters
pulse_width = 10e-9;  % Pulse width in seconds
pulse_amplitude = 1;  % pulse width
fs = 1e9;  % Sampling Rate (1 GHz)

% Generate the laser pulse
t = 0:1/fs:50e-9;  % Simulation time
laser_pulse = pulse_amplitude * rectpuls(t - 25e-9, pulse_width);

% Graph laser pulse
figure;
plot(t, laser_pulse);
title('LiDAR Laser Pulse');
xlabel('Time (s)');
ylabel('Amplitud');
```

b) Signal and Noise Reflection

We simulate the behavior of the reflected pulse when encountering an object and add noise to represent the real conditions of a LIDAR environment.

```
% Simulate pulse reflection
distance = 1500;  % Distance to object in meters
c = 3e8;  % Speed of light in m/s
```

```
delay_time = 2 * distance / c;  % Delay time by distance

% Create the reflected signal with delay and noise
reflected_signal = [zeros(1, round(delay_time * fs)), laser_pulse];
reflected_signal = reflected_signal + 0.05 * randn(1,
length(reflected_signal));  % Añadir ruido

% Plot reflected signal
figure;
plot(reflected_signal);
title('Reflected Signal with Noise');
xlabel('Samples');
ylabel('Amplitud');
```

c) Noise Filtering (Kalman Filter)

To improve the accuracy of LIDAR measurements, we implemented a **Kalman filter**, ideal for smoothing out noisy measurements.

```
% Kalman Filter Parameters
estimated_position = 0;  % Initial estimated position
velocity = 3;  % Constant speed of the object (in m/s)
P = 1;  % Initial estimated error
R = 0.05;  % Measurement noise
Q = 0.01;  % Process noise

% Initialization
estimated_signal = zeros(size(reflected_signal));

for k = 1:length(reflected_signal)
    % Prediction
    estimated_position = estimated_position + velocity * (1/fs);
    P = P + Q;

    % Update
    K = P / (P + R);
    estimated_position = estimated_position + K *
(reflected_signal(k) - estimated_position);
    P = (1 - K) * P;

    estimated_signal(k) = estimated_position;
end

% Plot filtered signal
figure;
plot(estimated_signal);
title('Signal Filtered with Kalman Filter');
xlabel('Samples');
```

```
ylabel('Amplitud');
```

d) Point Cloud Processing and Segmentation (with Lidar Toolbox)

If you have access to the **Lidar Toolbox**, you can process the distance data and generate a 3D point cloud. This is useful for visualization and analysis of detected objects

```
% Create a set of simulated points (X, Y, Z) for the point cloud
ptCloud = pointCloud([randn(100, 1), randn(100, 1), randn(100, 1) *
50]);

% View point cloud
figure;
pcshow(ptCloud);
title('LIDAR Point Cloud');
xlabel('X');
ylabel('Y');
zlabel('Z');
```

3. Implementation of Multiple Light Sources

To add redundancy, you can simulate multiple laser sources facing different directions and average them for greater measurement accuracy.

```
3. Implementation of Multiple Light Sources

To add redundancy, you can simulate multiple laser sources facing
different directions and average them for greater measurement accuracy.



% Generate multiple laser pulses

num_lasers = 3;  % Three light sources

multi_laser_signals = zeros(num_lasers, length(t));



for i = 1:num_lasers
```

```
    delay = randi([1, 100], 1);  % Random delay for each laser

    multi_laser_signals(i, :) = circshift(laser_pulse, delay);

end



% Average the signals from the different lasers

combined_signal = mean(multi_laser_signals, 1);



% Plot combined signals

figure;

plot(t, combined_signal);

title('Combined Signals from Multiple Light Sources');

xlabel('Time (s)');

ylabel('Amplitud');
```

4. Key Parameters for an Airborne LIDAR

To fit the system to an Airbus aircraft, some key parameters to consider include:

- **laser wavelength**: Generally between 905 nm and 1550 nm, depending on the application.
- **Sampling rate**: High to capture precise details during flight.
- **Maximum detection distance**: Depending on the altitude of the aircraft, but typically up to several kilometers.
- **Optical filters**: To reduce interference from the sun and other unwanted light sources.

5. Improvements for Series Production

- **Scalability**: Design the system so that laser sources and detectors can be easily replaced.
- **Weight and energy consumption optimization**: Uses low consumption and high efficiency components.
- **Real time processing**: Uses FPGAs or microcontrollers to process LIDAR data in real time and avoid delays in signal analysis.

Summary:

This design and code provides a foundation for an airborne LIDAR system with redundancy, noise filtering, and 3D data visualization. Use modules like **Signal Processing Toolbox** and the **Lidar Toolbox** allows you to simulate and process LIDAR signals efficiently.