- ## Mechanical

1 -

- **Functions**: This refers to the specific tasks and operations that the designed part or product must perform. The design should address these functions accurately and efficiently.

- **Safety:** Safety is a critical aspect of design, ensuring that the product doesn't pose risks or hazards to users, operators, or the environment during its operation and lifecycle.

- **Reliability:** A reliable design ensures that the part consistently performs its intended functions under various conditions over its expected lifespan, minimizing the chances of failures or malfunctions.

- **Manufacturability:** Design for manufacturability involves creating a design that can be produced using available manufacturing processes, materials, and techniques. This criterion aims to minimize production challenges, costs, and time.

- **Weight and Size:** The weight and size of a design can impact its performance, transportation costs, and ease of use. Balancing strength and functionality with weight and size considerations is essential.

- **Wear:** Consider the potential wear and tear that the part may experience during its operation. Design features that reduce friction, abrasion, and deterioration can improve the part's longevity.

- **Maintenance:** Design for ease of maintenance and repair. Parts that are designed with accessibility in mind, allowing for straightforward disassembly and replacement, can minimize downtime and maintenance costs.

- **Liability:** Liability refers to the legal responsibility a designer or manufacturer has if their product causes harm or damage. Design should account for potential risks, adhere to safety standards, and include appropriate warnings or instructions.
-

2 –

- Compression, Tension, Shear, Bending, Torsion, and Fatigue.

3 –

- Balance rotating parts.
- Use damping materials or devices.
- Isolate components with mounts or springs.
- Adjust natural frequencies to avoid resonance.
- Modify stiffness to change natural frequencies.
- Install active or passive damping devices.
- Align components properly.
- Minimize friction between moving parts.
- Use vibration analysis to identify issues.
- Strengthen structures to withstand vibrations.

4-

Harmonic Drive (Strain Wave Gear):

Advantages: Compact, precise, high torque, smooth motion, lightweight.

Planetary Gear System:

Advantages: Compact, high torque, efficient, versatile.

Cycloidal Drive:

Advantages: Compact, high torque, precise, strong.

Rack and Pinion:

Advantages: Linear motion, efficient, strong.

Worm Gear System:

Advantages: High gear reduction, self-locking, strong.

Spur Gear System:

Advantages: Simple, efficient, versatile.

- ## Electrical

1 –

Preferred Battery Type: LiPo (Lithium Polymer) batteries.

Why: LiPo batteries are lightweight, provide high energy, and offer strong power bursts, which suit the performance needs of remote vehicles.

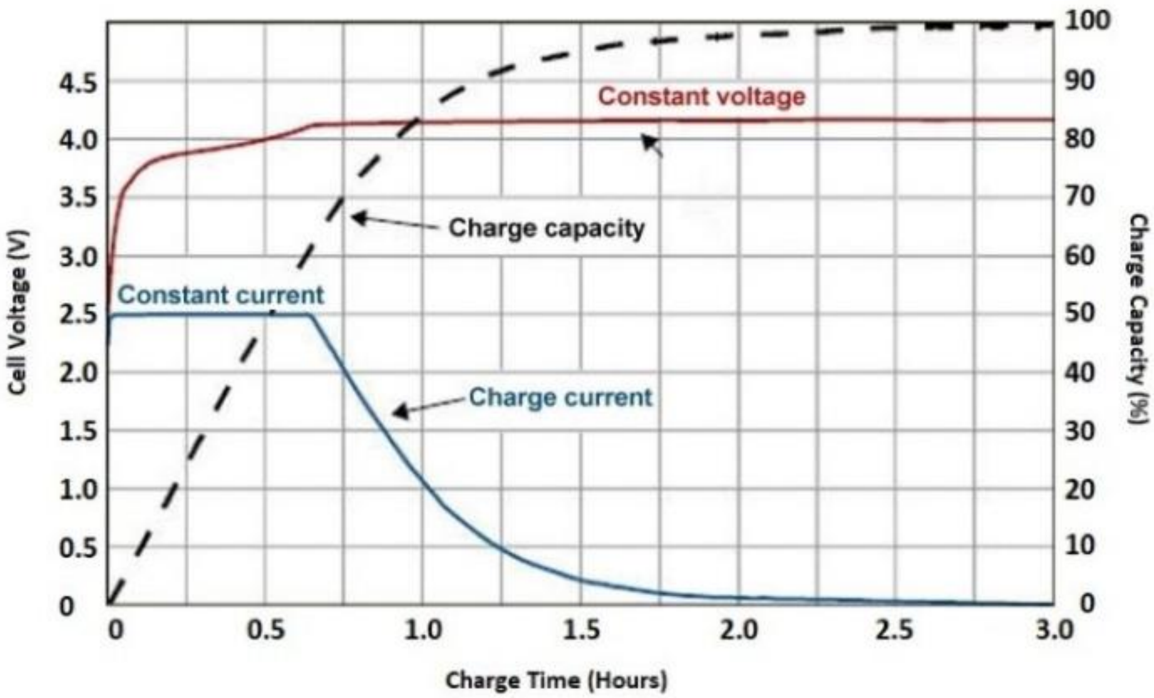
Choosing Battery Specifications:

- Voltage (Number of Cells): Match the battery's voltage to your vehicle's needs.
- Capacity (mAh): Choose capacity for desired runtime.
- Discharge Rate (C Rating): Match or exceed component current draw.
- Connector Type: Ensure compatibility.
- Size and Weight: Fit within vehicle, balance capacity and weight.
- Balance Charging: Use a charger that balances cells.
- Safety: Store and charge properly for safety.

2 –

The function of a BMS (Battery Management System) is to monitor, control, and protect a battery pack in various applications. Its primary goal is to ensure the safe and optimal operation of the battery by managing its charging, discharging, and overall health.

3 –



This charge curve of a Lithium-ion cell plots various parameters such as voltage, charging time, charging current and charged capacity. When the cells are assembled as a battery pack for an application, they must be charged using a constant current and constant voltage (CC-CV) method. Hence, a CC-CV charger is highly recommended for Lithium-ion batteries.

• The CC-CV method starts with constant charging while the battery pack's voltage rises.

• When the battery reaches its full charge cut-off voltage, constant voltage mode takes over, and there is a drop in the charging current.

• The charging current keeps coming down until it reaches below 0.05C.

• The battery reaches full charge voltage some time after the CV mode starts (as soon as one of the cells reaches its full charge voltage). At this stage, estimating SoC (state of charge) based on the battery voltage would mean that the battery is fully charged.

- The battery reaching its full charge voltage at this stage does not mean that it is 100% charged. Trickle charge mode kicks in immediately after this stage, where a reducing charging current charges the remaining battery capacity while balancing the cells at the same time.
- When every cell has been balanced and has reached its full charge voltage, at this point, the battery pack is really 100% charged. One way to know this is when the charging current has reached close to 0.05C.

4 –

The general electrical architecture of an Unmanned Aerial Vehicle (UAV), commonly known as a drone, consists of various components that work together to enable its operation and control. Here's an overview of the key components and their purposes:

Flight Controller: The brain that controls how the drone flies.

- Power Distribution Board (PDB): Sends power to all parts of the drone.
- Battery: Provides energy for flying.
- Motors and ESCs: Make the drone move by spinning the propellers.
- Flight Sensors: Help the drone stay stable and know where it's going.
- Radio Control System: Lets a pilot control the drone with a remote.
- Telemetry System: Sends data between the drone and the ground.
- Communication System: Lets the drone talk to the pilot and other drones.
- Onboard Computer: Thinks and makes decisions for advanced tasks.
- Camera and Sensors (optional): Collect information or take pictures.
- Safety Systems: Keep the drone and people safe in emergencies.

5 –

Minimizing electromagnetic interference (EMI) in printed circuit boards (PCBs) is crucial to ensure the proper functioning of electronic devices and to comply with electromagnetic compatibility (EMC) regulations. Here are some simplified strategies to reduce EMI in PCBs:

Keep Components Apart: Separate sensitive parts and group similar ones together.

- Solid Ground Plane: Use a good ground layer as a reference for signals.
- Match Traces: Make signal paths equal in length to avoid confusion.
- Use Capacitors: Add capacitors near power pins to clean up noise.
- Shielding and Covers: Use shields or cases to keep signals contained.
- Pick Components Wisely: Choose parts that naturally reduce noise.
- Add Filters: Put small parts to filter noise in signal and power lines.
- Be Careful with Clocks: Handle clocks well to avoid noise problems.
- Get Expert Advice: Ask experienced engineers for help if needed.
- Follow Rules: Stick to standards and guidelines for good design.

Perception

1 –

Neural Network:

A neural network is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes, or "neurons," organized in layers. Each neuron processes information and passes it to the next layer, allowing the network to learn patterns and make predictions from data.

Types:

- Feedforward (Simple pattern recognition)
- Convolutional (Recognizing images)
- Recurrent (Understanding sequences)
- Generative (Creating new things)

Applications:

- Images: Identifying objects in pictures.
- Language: Translating languages, understanding text.
- Speech: Converting spoken words to text.
- Cars: Helping cars drive on their own.
- Health: Diagnosing diseases from medical images.
- Finance: Predicting stock prices and finding fraud.
- Games: Making game characters smart.
- Art: Creating music, art, and more.

- Industry: Making factories work better.
- Science: Solving complex problems and analyzing data.
- Neural networks help computers do many smart things by learning from examples, just like we learn from experience.

2 –

```
import numpy as np

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers


# Load the Fashion MNIST dataset

(x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data()


# Preprocess the data

x_train = x_train / 255.0

x_test = x_test / 255.0


# Build the neural network architecture

model = keras.Sequential([

    layers.Flatten(input_shape=(28, 28)),  # Flatten the 28x28 images

    layers.Dense(128, activation='relu'),   # Hidden layer with 128 units

    layers.Dropout(0.2),                    # Dropout for regularization

    layers.Dense(10, activation='softmax')  # Output layer with 10 classes

])


# Compile the model

model.compile(optimizer='adam',

              loss='sparse_categorical_crossentropy',

              metrics=['accuracy'])
```

```
# Train the model
model.fit(x_train, y_train, epochs=10, validation_split=0.2)

# Evaluate the model on test data
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"Test accuracy: {test_acc}")
```

3 –

```
import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Load a small subset of the ImageNet dataset
# For the complete ImageNet dataset, consider using TensorFlow Datasets or other sources.
# Here, we assume you have a directory structure with train and validation folders.
train_data_dir = 'path_to_train_data'
validation_data_dir = 'path_to_validation_data'

# Preprocess images and apply data augmentation
train_datagen = ImageDataGenerator(
   rescale=1.0/255.0,
   rotation_range=20,
   width_shift_range=0.2,
   height_shift_range=0.2,
   shear_range=0.2,
   zoom_range=0.2,
   horizontal_flip=True,
   fill_mode='nearest')
```

```
validation_datagen = ImageDataGenerator(rescale=1.0/255.0)

# Load pretrained ResNet50 model
base_model = ResNet50(include_top=False, weights='imagenet')

# Add custom classification head
model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(1000, activation='softmax')
])

# Compile the model
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Load and preprocess data using the data generators
train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical')

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
```

```
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical')

# Train the model
model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_size)
```

4 –

LiDAR (measuring distances with lasers), RGB cameras (taking color pictures), and depth cameras (measuring depth in images).

Working Principle:

Combine data from these sensors to find and understand objects in the world:

Collect Data: Sensors gather information about the environment.

Combine Data: Mix data from different sensors to understand what's around.

Analyze Features: Look at colors, shapes, and more to figure out what objects are there.

Find Objects: Use smart methods to detect and locate objects.

Know Where: Understand where objects are in 3D space.

Improvements:

Better Mixing: Improve how sensors' data are combined.

Smarter Algorithms: Develop better computer programs to understand the data.

Understand Context: Teach computers to understand the scene better.

Use More Data: Make models smarter by using more examples.

Faster Processing: Make everything work faster for real-time use.

Better Sensors: Use newer, more accurate sensors.

5 –

Stereo-Camera:

Has two cameras for depth perception.

Takes two pictures to understand distance.

Good for 3D tasks like mapping.

Mono-Camera:

Has one camera, no depth perception.

Used for regular pictures.

Transforming 2D to 3D (Stereo Camera):

Use two images and find the difference between their positions (disparity).

Use the disparity to calculate depth.

Convert the pixel coordinates to world coordinates using equations.