Ahmed Ayman Samir

ID:2000069 | AIN SHAMS UNIVERSITY|MY RESUME

AI SOLO MISSION.

Questions:-

PRECEPTION

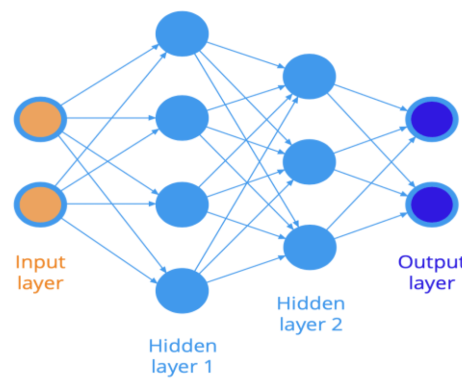
1- **What is a neural network? What are its types and applications?**

Definition:

A neural network is **a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates**. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. It also contains input layer ,hidden layer 1 &2 ,output layer

Types:

- Artificial Neural Networks (ANN)
- Feed Forward artificial neural networks
- Perceptron and multilayer perceptron neural networks
- Modular neural networks
- Recurrent neural networks
- Convolution Neural Networks (CNN)
- Recurrent Neural Networks (RNN)



Applications:

1. Facial Recognition

Facial Recognition Systems are serving as robust systems of surveillance. Recognition Systems matches the human face and compares it with the digital images. They are used in offices for selective entries. The systems thus authenticate a human face and match it up with the list of IDs that are present in its database.

Speech recognition

Speech recognition has many applications such as virtual assistance , hands-free computing and video games ,etc neural networks is used widely in this areas .

Character recognition

In this Application of AI is also a use of neural networks in character recognition, where this technology is used to recognize a character with the help of this application of AI.

Spell Checking

Personalized Spell Checking using Neural Networks a new system for detecting misspelled words was proposed. this system corrects the mistakes made by typist.

2- **Write Python code that compiles and trains a deep neural network on the Fashion MNIST dataset. Assume and tune your parameters and hyper-parameters?**

All details are added on GitHub ➡ Here

3- **Write Python code that compiles and trains a deep neural network for the ImageNet dataset. Assume and tune your parameters and hyper-parameters?**

All details are added on GitHub ➡ Here

4- **Read the LaserNet paper used for 3D-object detection. What sensors does it require? What is the main working principle of this technique? Do you see any way to improve it?**

Sensors to detect 3D-objects are:

LIDAR sensor: has the ability to express 3D spatial information

proximity sensors

sensor technologies like ultrasonic sensors

LiDAR range sensors are commonly used for this task because they generate accurate range measurements of the objects of interest independent of lighting conditions. To be used in a real-time autonomous system, it is important that these approaches run efficiently in addition to having high accuracy. Also, within the context of a full selfdriving system, it is beneficial to have an understanding of the detector's uncertainty. LiDAR range sensors used on autonomous vehicles employ a variety of physical approaches to point one or more range-measuring lasers in different directions (physically spinning, raster scanning, MEMS mirrors, etc),

Ways of improvements:

1. Quickly process point clouds and integrate them with other data sources.
2. Use LiDAR for 3D city modelling.
3. Make a digital terrain and surface model from a point cloud.
4. Integrate programs like LAStools into your workflows.

5- What is the difference between stereo-camera and mono-camera? State mathematically how coordinates can be transformed from the 2D pixel domain of the two images of the stereo-camera to 3D world coordinates.

Stereo-camera	Mono-camera
A type of camera with two or more lenses with a separate image sensor or film frame for each lens. This allows the camera to simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images, a process known as stereo photography. Stereo cameras may be used for making stereoviews and 3D pictures for movies, or for range imaging. The distance between the lenses in a typical stereo camera (the intra-axial distance) is about the distance between one's eyes (known as the intra-ocular distance) and is about 6.35 cm, though a longer base line (greater inter-camera distance) produces more extreme 3-dimensionality.A camera having two matched lenses separated about the same distance as a person's eyes so that two pictures to be viewed in a stereoscope or projected to give a stereoscopic impression can be taken simultaneously.. -- It allows the camera to simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images, a process known as stereo photography. -- **provide a different image to each eye**	A single camera sensor mounted and capturing video that needs to be processed and analyzed, that system is called a monocular (single- eyed) system where each position on an image can record and show a different *amount* of light, but not a different hue. It includes all forms of black-and-white photography, which produce images containing shades of neutral grey ranging from black to white.[1] Other hues besides grey, such as sepia, cyan, blue, or brown can also be used in monochrome photography.[2] In the contemporary world, monochrome photography is mostly used for artistic purposes and certain technical imaging applications, rather than for visually accurate reproduction of scenes. -- **a common type of vision sensor used in automated driving applications**. When mounted on an ego vehicle, this camera can detect objects, detect lane boundaries, and track objects through a scene. Before you can use the camera, you must calibrate it. -- **Provide** Unlike color cameras, monochrome cameras **capture all the incoming light at each pixel – irrespective of color**. Since red, green, and blue are all absorbed simultaneously, each pixel can receive up to three times the amount of light.

FIG .1

FIG . 2

FIG.3

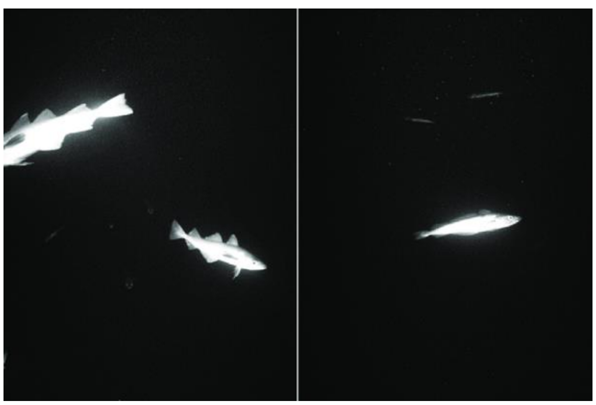


FIG .4

As a beginning for converting 2D to 3D:-

Finding the 2D pixel coordinates of a 3D Point Explained from Beginning to End When a point or vertex is defined in the scene and is visible to the eye or to the camera, it appears in the image as a dot (or more precisely a pixel if the image is a digital one). We already talked about the perspective projection process which is used to convert the position of that point in 3D space to a position on the surface of the image. But this position is not expressed in terms of pixels coordinates. So how do we actually find the final 2D pixel coordinates of the projected point in the image? In this chapter, we will review the entire process by which points are converted from their original world position to their final raster position (their position in the image in terms of pixel coordinates). If co-ordinates are given in parametric form x(t),y(t) forming any base contour, then **add z=cu, where c is chosen for the depth you want**. (x(t),y(t),cu); In general conversion of 2D projection to 3D as you ask is indeterminate

For more info about the mathematical relations I used this pdf

REFRENCE;

6- What are the different approaches to odometry sources and localization in self-driving cars?

First of all : Self driving cars Technology A self-driving car (sometimes called an autonomous car or driverless car) is a vehicle that **uses a combination of sensors, cameras, radar and artificial intelligence (AI)** to travel between destinations without a human operator.

Visual odometry is **a method to estimate the pose by examining the changes that motion induces in the onboard camera**. Previously, we extracted features f[k - 1] and f[k] from two consecutive frames I[k - 1] and I[k] . We can use these features to estimate the camera motion from 3D-2D point correspondences.

Localization is **a critical capability for autonomous vehicles, making it possible to pinpoint their location within centimeters inside a map**. This high level of accuracy enables a self-driving car to understand its surroundings and establish a sense of the road and lane structures

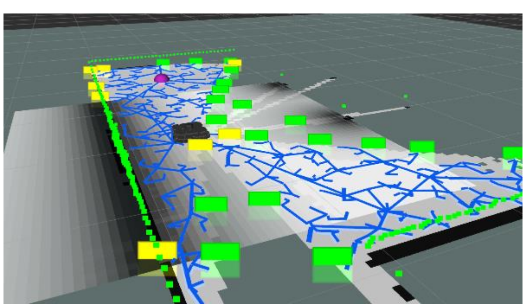
Their so many types to make the car locate itself as

SLAM —A very popular technique if we also want to estimate the map exists. It is called SLAM (Simultaneous Localization and Mapping). In this technique, we estimate our position but also the position of landmarks. A traffic light can be a landmark
Inertial Measurement Unit (IMU) is a sensor capable of defining the movement of the vehicle along the yaw, pitch, roll axis.

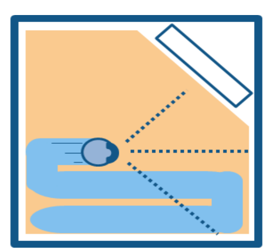
7- What is known by SLAM? What are its different types?

SLAM (simultaneous localization and mapping) is **a method used for autonomous vehicles that lets you build a map and localize your vehicle in that map at the same time**.
Consider a home robot vacuum. Without SLAM, it will just move randomly within a room and may not be able to clean the entire floor surface. In addition, this approach uses excessive power, so the battery will run out more quickly. On the other hand, robots with SLAM can use information such as the number of wheel revolutions and data from cameras and other imaging sensors to determine the amount of movement needed. This is called localization. The robot can also simultaneously use the camera and other sensors to create a map of the obstacles in its surroundings and avoid cleaning the same area twice. This is called mapping.

The widely *TYPES* uses SLAM sensors are the Global Positioning System (GPS), Parallel Tracking and Mapping (PTAM) • LSD-SLAM (available as open-source) • S-PTAM (available as open-source) • ORB-SLAM (available as open-source) • ORB-SLAM2 (available as open-source)rotary encoders, infrared (IR), acoustic sensors (ultrasound, microphone, and sonar sensor), camera, inertial measurement unit (IMU), ultrawideband (UWB), acoustic, LiDAR, RADAR, and RGB-D sensors.



Without SLAM:
Cleaning a room randomly.



With SLAM:
Cleaning while understanding the room's layout.

8- Describe briefly the ICP algorithm used for localization using LIDAR point clouds. Write a pseudo-code for the algorithm

Iterative closest point (ICP) is **an algorithm employed to minimize the difference between two clouds of points** also PCL (Point Cloud Library) is **an open-source framework for n-dimensional point clouds and 3D geometry processing**. It includes several variants of the ICP algorithm. Open source C++ implementations of the ICP algorithm are available in VTK, ITK and Open3D libraries.

Reference

PSEUDO-CODE:-

ICP (point set P(source scan),point set Q(Target scan))

Compute (centre of mass for p and q){}

Translate(centre of mass for Q into P){}

While (d(T).Emax||Number of iteration > for examples 20)

{ for each pi in P{

Si = nearest point(pi,Q)

}

Transformation T = min t E(T) = min t Σ ||qi – T(pi)||

P = Transformation point set (P,T)

Number of iterations++

}

There are several codes on GitHub Reference It is written On c++ language

9- What are the types or variants of the Kalman filter? What are the uses and applications?

A wide variety of Kalman filters exists by now, from Kalman's original formulation - now termed the **"simple" Kalman filter, the Kalman–Bucy filter, Schmidt's "extended" filter, the information filter, and a variety of "square-root" filters** that were developed by Bierman, Thornton, and many others.

Types

- Linear Kalman filter.
- Extended Kalman filter.
- Unscented Kalman filter.

Nearly all the real-time operations are non-linear and all the techniques can be described as a discrete-time system to a remarkable extent of precision utilizing very small time steps. Now the concern is to evaluate the states of this discrete-time controlled function and the procedure is typically described with the use of a linear stochastic difference equation. This computation can be efficiently and accurately done by the Kalman Filters

USES &Applications

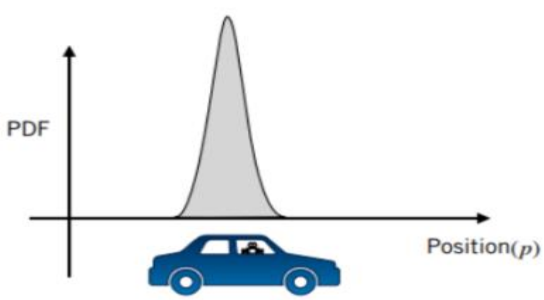
A common application is for guidance, navigation, and control of vehicles, particularly aircraft, spacecraft and ships positioned dynamically. Furthermore, Kalman filtering is a concept much applied in time series analysis used for topics such as signal processing and econometrics.

It is used in many applications as Finance, radar and image processing

Tracking objects (e.g., missiles, faces, heads, hands)

- Fitting Bezier patches to (noisy, moving, ...) point data
- Economics
- Navigation
- Many computer vision applications – Stabilizing depth measurements – Feature tracking – Cluster tracking – Fusing data from radar, laser scanner and stereo-cameras for depth and velocity measurements

10- Write C++ code that implements the linear Kalman filter for one predict and one update step for the following problem. (Hint: You may use Eigen library)



$$\mathbf{x} = \begin{bmatrix} p \\ \frac{dp}{dt} = \dot{p} \end{bmatrix} \qquad \mathbf{u} = a = \frac{d^2p}{dt^2}$$

Motion/Process Model

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

Position Observation

$$y_k = [1 \quad 0]\, \mathbf{x}_k + v_k$$

Noise Densities

$$v_k \sim \mathcal{N}(0,\, 0.05) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\, (0.1)\mathbf{1}_{2\times2})$$

Can be found here on Github [Here]

Path Planning

11- What are the different path planning algorithms that are used in autonomous mobile robots?

Algorithms of global path planning are mainly divided into two types: **heuristic search methods and intelligent algorithms**. The initial representation of the heuristic search is the A∗ algorithm developed by the Dijkstra algorithm.

The survey shows **GA (genetic algorithm), PSO (particle swarm optimization algorithm), APF (artificial potential field), and ACO (ant colony optimization algorithm)** are the most used approaches to solve the path planning of mobile robot.

Supervised learning algorithms include **MSVM, LSTM, MCTS and CNN**. Optimal value reinforcement learning algorithms include Q learning, DQN, double DQN, dueling DQN. Policy gradient algorithms include policy gradient method, actor-critic algorithm, A3C, A2C, DPG, DDPG, TRPO and PPO

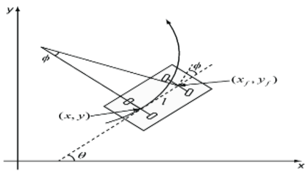
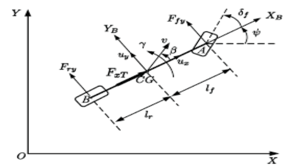
12- Write Python code that implements BFS (Breadth First Search) or A* Algorithm - whichever you find yourself comfortable with - algorithm on a 5 x 5 occupancy grid. Obstacles on (2,0), (2,1), (2,2) Start on (1,1) - Goal on (4,1) Assume the grid is a 2D NumPy array, obstacle cells have a value of 1, and free cells have a value of 0.

The Solution is on Github ➡ Here (Not Sure But I did My Best)

Navigation Control

13- What is the difference between kinematic and dynamic modeling of vehicles?

Kinematic Model	Dynamic model
Kinematics is the study of motion without regard to forces that cause it	**Dynamics is the study of motions that result from forces**.
(x, y) are the coordinates of the rear axle midpoint, θ is the orientation of car, ϕ is the steering angle and l is the wheelbase. Source publication. A vehicle model for micro-traffic simulation in dynamic urban scenarios. Conference Paper.	The most simplified vehicle dynamic model is **a two-degree-of-freedom bicycle model, representing the lateral and yaw motions**. The idea behind this model is that sometimes it is not necessary or desirable to include the longitudinal direction, because it does not affect the lateral or yaw stability of the vehicle.
There are **two types** of kinematics: particle kinematics and rigid body kinematics. Particle kinematics is the study of the motion of a single particle, while rigid body kinematics is the study of the motion of an object that doesn't deform over time.	The vehicle driving dynamics is divided into **longitudinal dynamics and vertical dynamics**, which includes driving, braking and ride comfort. The problem of driving slip and braking slip are solved by the study of vehicle longitudinal tire force, which can also improve driving and braking efficiency.

14- State different longitudinal and lateral vehicle controllers.

General difference

The longitudinal controller is responsible for regulating the vehicle's cruise velocity while the lateral controller steers the vehicle's wheels for path tracking

Also:

The term "longitudinal controller" is typically used in referring to **any control system that controls the longitudinal motion of the vehicle**, for example, its longitudinal velocity, acceleration or its longitudinal distance from another preceding vehicle in the same lane on the highway.

Laterial Vehicle control is the final step in a navigation system and is typically accomplished using two independent controllers: Lateral Controller: **Adjust the steering angle such that the vehicle follows the reference path**. The controller minimizes the distance between the current vehicle position and the reference path.

15- Write C++ code that implements the forward kinematics of a kinematic bicycle model. Inputs = [v, δ] (velocity of the vehicle and steering angle) Outputs = [x, y, θ] (The 2D pose of the bicycle, x, y, and yaw) Assume your step-time for integration and any other missing parameters.

In this notebook, you will implement the kinematic bicycle model. The model accepts velocity and steering rate inputs and steps through the bicycle kinematic equations. Once the model is implemented, you will provide a set of inputs to drive the bicycle in a figure 8 trajectory.

The bicycle kinematics are governed by the following set of equations:

$$
\begin{aligned}
\dot{x}_c &= v\cos(\theta+\beta) \\
\dot{y}_c &= v\sin(\theta+\beta) \\
\dot{\theta} &= \frac{v\cos\beta\tan\delta}{L} \\
\dot{\delta} &= \omega \\
\beta &= \tan^{-1}\left(\frac{l_r\tan\delta}{L}\right)
\end{aligned}
$$

where the inputs are the bicycle speed v and steering angle rate ω. The input can also directly be the steering angle δ rather than its rate in the simplified case. The Python model will allow us both implementations.

In order to create this model, it's a good idea to make use of Python class objects. This allows us to store the state variables as well as make functions for implementing the bicycle kinematics.

The bicycle begins with zero initial conditions, has a maximum turning rate of 1.22 rad/s, a wheelbase length of 2m, and a length of 1.2m to its center of mass from the rear axle.

From these conditions, we initialize the Python class as follows:

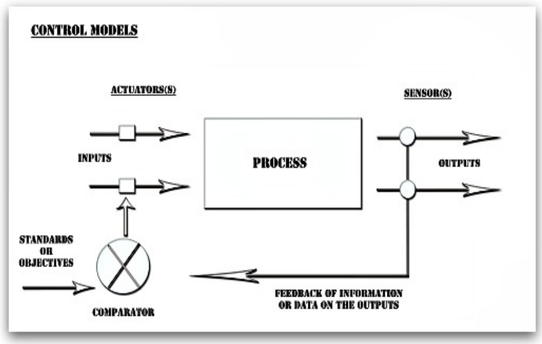
The Solution is on Github → **Here**

16- Why is modeling an important part of control (Ex: what could the bicycle model be used for in control)? What if there is no model, can you still control a system?

Modelling is for the process that would be controlled. It helps in understanding how the process would behave in various conditions.

Importance Because

improves process communication, increases control and consistency, aligns operations with business strategy, improves operational efficiencies.



There are so many types of modeling as:-

Calculating model, graphical model,

Mathematical modeling of a control system is **the process of drawing the block diagrams for these types of systems in order to determine their performance and transfer functions**.

Graphical Model **gives graphical representation of system to be built**. Modeling contributes to a successful software organization. Modeling is a proven and well accepted engineering technique.

IF THERE IS NO MODEL A SYSTEM WILL BE HARD TO BE CONTROLLED

17- What is the MPC (model predictive control), how does it work? What are the most common issues with it? Low-Level Control

Model predictive control (MPC) is **an advanced method of process control that is used to control a process while satisfying a set of constraints**. It has been in use in the process industries in chemical plants and oil refineries

How it works: MPC **uses a model of the plant to make predictions about future plant outputs**. It solves an optimization problem at each time step in order to find the control action that drives the predicted plant output as close to the desired reference as possible.

More Details ➡ Here

Low-Level Control

18- Perform closed loop control on a steering wheel in a self-driving car. Define your controller, the final control element, and the process variable. (Draw the block diagram).

\\

19- What are the hardware and actuators needed for low-level control in self-driving cars?

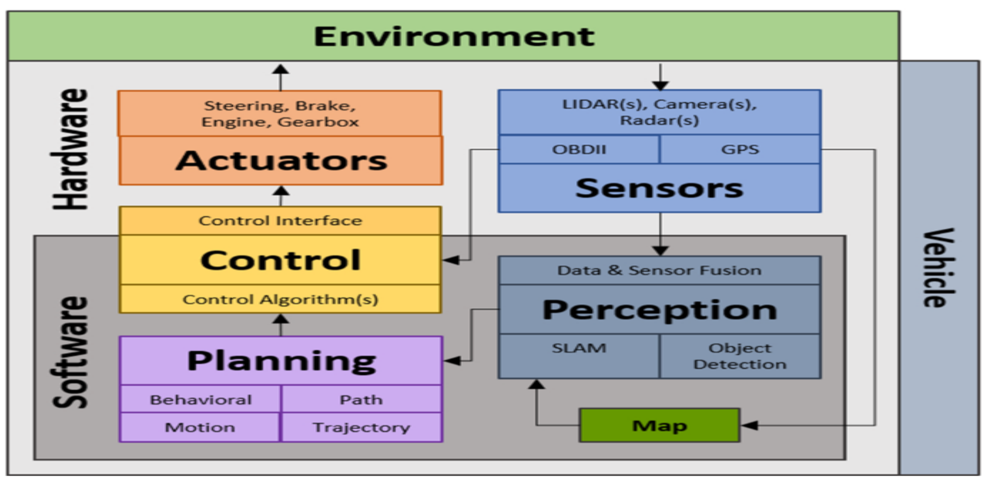
Hard Ware

Camera, Radar, Lidar, GPS is most commonly used sensor for a self-driving car. Besides, 16-line laser radar, single-line laser radars, millimeter-wave radars, cameras, and a GPS unit make up a common sensor setup for self-driving cars.

Actuators

The operation of an autonomous vehicle requires actuation of 3 major controls:

- Acceleration – Throttle Actuation.
- Direction – Steering Actuation.
- Stopping – Brake Actuation.
- Raining and dirt Actuation
- Tires Auto filling air Sensor



My GitHub Account

My Resume Google Drive Link

REFRENCES:- Stack Over Flow **Website**

GitHub **Website**

GeekForGeeks **Website**

Note:- Although the internet was in trouble at home, I managed to find a solution and finish the task