**Objectives**

I am fascinated by how system such as tesla and other automated systems like drones work. I like how they are able to get information from their surrouinding, process the information and act based on those information. Now I will want to focus my attention onto a self-driving car, a fully autonomous self-driving car, although not with a production car as that will be way too complex.

Questions to answer – What level of autonomy am I aiming for: As I said above, I am aiming for fully autonomous

Define requeirements (So what functionalities and features do you want your self-driving car to have): Some of the features includes:

1. Ability to compute route given destination
2. Ability to compute shortest path to final destination
3. Ability to go from point A to point B without human interference
4. Ability to create 3D mapping of its position

**What is a self-driving car**

A self-driving car (sometime called an autonomous car or driveless car) is a vehicle that uses a combincation of sensors, cameras, radar and AI to travel between destinations without a human operator. To qualify as fully autonomous, a vehicle must be able to navigate without human intervention to a predetermined destination over roads that have not been adapted for its use.

A close up of a text

Description automatically generated

**How self-driving cars work**

A screenshot of a computer

Description automatically generated

A white text on a white background

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a car

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a web page

Description automatically generated

[What are Self-Driving Cars and How Do They Work? (techtarget.com)](https://www.techtarget.com/searchenterpriseai/definition/driverless-car#:~:text=A%20self-driving%20car%20%28sometimes%20called%20an%20autonomous%20car,to%20travel%20between%20destinations%20without%20a%20human%20operator.)

**What is AI**

**Image recognition system**

**Machine learning**

**Neural networks**

Creating an autonomous system like a self-driving car is indeed a complex and multi-disciplinary task that involves various fields such as robotics, artificial intelligence, computer vision, sensor fusion, control theory, and more. Here's a high-level step-by-step guide to help you get started:

1. Define the Objectives and Requirements:
   * Determine the scope of your project: What level of autonomy are you aiming for? Is it a fully autonomous vehicle or a semi-autonomous one?
   * Define the requirements: What functionalities and features do you want your self-driving car to have? What are the safety and regulatory requirements?
2. Research and Study:
   * Familiarize yourself with the basics of robotics, artificial intelligence, computer vision, sensor technologies, and control systems.
   * Study existing self-driving car technologies, algorithms, and platforms. This will help you understand the challenges and solutions in this field.
3. Choose a Platform:
   * Decide whether you want to build your self-driving car from scratch or use an existing platform.
   * If you're starting from scratch, consider platforms like ROS (Robot Operating System) for robotics development.
4. Design the Hardware:
   * Select appropriate sensors such as LiDAR, cameras, RADAR, GPS, IMU (Inertial Measurement Unit), etc., based on your requirements.
   * Choose the computing platform (e.g., NVIDIA Jetson, Raspberry Pi) for processing sensor data and running algorithms.
5. Develop Perception Algorithms:
   * Implement computer vision algorithms for object detection, recognition, and tracking using cameras.
   * Use LiDAR and RADAR data for 3D mapping, localization, and obstacle detection.
   * Fuse sensor data using techniques like sensor fusion to improve accuracy and reliability.
6. Implement Decision-Making Algorithms:
   * Develop algorithms for path planning and trajectory generation to navigate the vehicle safely and efficiently.
   * Design decision-making algorithms for behavior planning and control, considering factors like traffic laws, road conditions, and vehicle dynamics.
7. Integrate and Test:
   * Integrate hardware components, sensors, and software modules into a cohesive system.
   * Conduct extensive testing in simulation environments and real-world scenarios to validate the performance and safety of your self-driving car.
8. Iterate and Refine:
   * Continuously iterate on your design, algorithms, and system implementation based on testing results and feedback.
   * Address any issues or challenges encountered during development and testing.
9. Ensure Safety and Compliance:
   * Ensure that your self-driving car meets safety standards and regulatory requirements.
   * Implement fail-safe mechanisms and redundancy to handle unexpected situations or failures.
10. Deployment and Deployment:

* Once your self-driving car meets the desired level of performance, safety, and reliability, deploy it in real-world scenarios or controlled environments for further validation and refinement.

Keep in mind that building a self-driving car is a challenging and time-consuming process that requires expertise in various domains. Collaboration with experts in robotics, AI, and automotive engineering can be beneficial in tackling the complexities of the project. Additionally, staying updated with the latest advancements in autonomous driving technology and research can help you make informed decisions throughout the development process.

Top of Form

**Learning robotics**

Learning about robotics from scratch is a vast and exciting journey that encompasses various disciplines such as mechanical engineering, electrical engineering, computer science, and control theory. While it's not possible to cover everything in a single response, I can provide you with a structured approach to get started:

1. **Basics of Robotics**:
   * Understand what robotics is and its applications in different fields.
   * Learn about the main components of a robot: sensors, actuators, controllers, and the mechanical structure.
   * Explore different types of robots such as industrial robots, mobile robots, humanoid robots, etc.
2. **Mechanical Engineering**:
   * Study basic mechanical principles such as kinematics, dynamics, and mechanics of materials.
   * Learn about mechanisms and linkages used in robot design.
   * Familiarize yourself with CAD (Computer-Aided Design) software for designing robot components.
3. **Electrical Engineering**:
   * Learn about basic electronics including circuits, voltage, current, and resistors.
   * Understand the principles of electromagnetism and how motors work.
   * Study sensors and actuators commonly used in robotics such as encoders, accelerometers, motors, and servos.
4. **Computer Science**:
   * Learn programming languages commonly used in robotics such as Python, C/C++, and MATLAB.
   * Study algorithms and data structures as they are essential for robot perception, decision-making, and control.
   * Gain familiarity with software development tools and libraries used in robotics such as ROS (Robot Operating System).
5. **Control Theory**:
   * Understand control systems and feedback loops.
   * Study classical control theory concepts such as PID (Proportional-Integral-Derivative) controllers.
   * Learn about modern control techniques including state-space representation and optimal control.
6. **Robotics Projects and Hands-On Experience**:
   * Start with simple robotics projects such as building a line-following robot or a robot arm controlled by Arduino.
   * Gradually increase the complexity of your projects as you gain more knowledge and experience.
   * Document your projects and keep a portfolio to showcase your skills and progress.
7. **Online Courses and Resources**:
   * Take advantage of online courses and tutorials available on platforms like Coursera, edX, Udemy, and Khan Academy.
   * Join robotics forums and communities to connect with other enthusiasts and experts in the field.
   * Watch YouTube videos and read books on robotics to supplement your learning.
8. **Hands-On Workshops and Hackathons**:
   * Participate in robotics workshops, hackathons, and maker events to gain practical experience and network with other enthusiasts.
   * Collaborate with others on robotics projects to learn from their expertise and perspectives.

Remember that learning about robotics is a gradual process, and it's okay to start with the basics and gradually build your skills and knowledge over time. Don't be afraid to experiment, make mistakes, and learn from them. Most importantly, stay curious and passionate about the field, and don't hesitate to seek help and guidance from others along the way