A

GPS (Global Positioning System):

1) GPS is a satellite-based navigation system: a constellation of satellites transmits signals that are picked up by receivers on or near Earth.
2) A receiver calculates its position by trilateration: measuring distance from multiple satellites, correcting for time/bias errors, and solving for x, y, z coordinates plus timing offset.
3) More broadly, GNSS is a general term for multiple global navigation satellite systems. A GNSS receiver can use signals from multiple constellations to improve accuracy and robustness.
4) It receives signals from at least 3 satellites to get the geological locations. A fourth satellite is needed to check if the signals are correct or not for accuracy.

Works nearly anywhere (especially outdoors) and in almost any weather.Accuracy depends on factors like satellite geometry, signal blockage or reflections, atmospheric delays, and clock errors.Vulnerable to jamming, spoofing, or signal blockage where direct satellite visibility is lost.

Useful applications:

1) Navigation (cars, ships, aircraft)

2) Asset or fleet tracking (vehicles, shipping containers, drones)

3) Mapping, surveying, geolocation services

4) Time synchronization (telecommunications, finance)

5) Location-based services (mobile apps, GIS)

IMU (Inertial Measurement Unit):

An IMU is a sensor device that measures linear acceleration (via accelerometers) and angular velocity (via gyroscopes), sometimes also magnetic fields (with a magnetometer).

For example:

1) 6-axis IMU = 3-axis accelerometer + 3-axis gyroscope
2) 9-axis IMU = 6-axis + 3-axis magnetometer

These sensors together track motion and orientation in 3D space.

IMUs complement GNSS by providing continuous motion data when satellite signals are weak, blocked, or lost.

1) GNSS gives absolute position (latitude, longitude, altitude).
2) IMU gives relative motion (acceleration, rotation). When fused via algorithms like the Kalman Filter, IMU, and GNSS together provide accurate, smooth, and reliable positioning even during GNSS outages.
3) **Dead Reckoning:** Estimating current position from last known data using IMU readings during GNSS loss.
4) **Sensor Fusion:** Combining accelerometer, gyroscope, and GNSS data to reduce drift and noise.
5) **Kalman Filter:** A Mathematical method that continuously corrects IMU errors using GNSS updates.

Useful Applications:

1) Vehicle tracking and navigation (autonomous cars, off-road, tunnels)

2) Collision and motion analysis

3) Flight testing and robotics

4) Any system requiring precise movement data under GNSS-denied conditions

Intel RealSense Depth Sensor:

The Intel RealSense is an RGB-D (color + depth) camera that captures both color and distance information. It uses stereo vision with two infrared (IR) cameras to compare their images to calculate depth and an IR dot projector that adds texture to flat surfaces, improving accuracy. All depth processing happens on the device, reducing CPU/GPU load on the host computer.

While GNSS provides global positioning, sensors like RealSense provide local 3D perception. In robotics or autonomous systems, RGB-D data from cameras like RealSense is fused with IMU and GNSS data to achieve complete navigation GNSS for global location, IMU for motion continuity, and depth sensing for environment awareness and obstacle detection.

Stereo Vision: Estimates depth by matching features between two cameras.

IR Dot Projection: Adds artificial texture for reliable depth on featureless surfaces.

Onboard Processing: Computes depth internally for real-time use.

RGB-D Data in AI: Depth maps can be added as a fourth channel (R, G, B, D) in CNNs to boost performance in segmentation and detection.

Useful Applications:

1) Robotics & Autonomous Vehicles: Object detection, obstacle avoidance, mapping.
2) AR/VR: Environment scanning and interaction.
3) Computer Vision Research: RGB-D datasets for deep learning.
4) Industrial & Safety Systems: Distance measurement, collision prevention.

LiDAR (Light Detection And Ranging):

LiDAR is an *active* remote sensing technology: it emits laser light pulses, which bounce off surfaces and return to the sensor. The time-of-flight of the pulse is used to compute distance. Because each pulse may reflect off multiple objects (tree branches, leaves, ground), you can get multiple “returns” per pulse.

The resulting data is often represented as a **point cloud** (a set of 3D points with x, y, z coordinates), sometimes along with additional attributes like intensity or classification (ground, vegetation, buildings, etc.).
Some LiDAR systems record the full waveform (the energy distribution of the return), while others record discrete returns (peaks) only.

Often LiDAR systems integrate other sensors (GPS, IMU) to know the position and orientation of the sensor when pulses were emitted.

Useful applications:

1) Mapping terrain, digital elevation models (DEM)
2) Forestry and ecological studies
3) Urban modeling (buildings, infrastructure)
4) Autonomous vehicles (for environment perception, obstacle detection)
5) Robotics (navigation, SLAM when combined with other sensors)
6) Surveying and infrastructure inspection

Wheel Odometry:

Wheel odometers measure wheel rotation to estimate the distance and direction of a robot’s movement. Although simple and long-used in robotics, wheel odometry alone produces noisy and error-prone data due to skidding, uneven terrain, and wheel wear.

SLAMcore enhances odometry by tightly integrating it with visual and inertial sensors (IMU and Intel RealSense D435i). Instead of loosely combining or averaging data, SLAMcore’s sensor fusion algorithm intelligently blends all three data sources, improving accuracy, robustness, and reliability in Simultaneous Localization and Mapping (SLAM).

Key Advantages:

1) **Error Compensation:** Strengths of each sensor offset weaknesses of others.

2) **Adaptability:** Odometry is valuable when visual data is unreliable (e.g., motion-heavy scenes).

3) **Faster Estimation:** Stationary wheel data constrains SLAM's search space, speeding computation.

4) **Simplified Calibration:** Multi-sensor fusion allows **automatic, one-pass calibration**, up to **10× faster** than traditional methods.

5) **Cost Efficiency:** Offers a **flexible and affordable** alternative to expensive LiDAR-based systems.

Useful Applications:

1) Mobile robots and autonomous systems
2) Consumer electronics requiring spatial awareness
3) Environments where external sensing (vision/LiDAR) is limited or unreliable

By fusing wheel odometry, IMU, and visual depth data, SLAMcore delivers a commercial-grade SLAM system that is more accurate, faster, and easier to deploy. The integration ensures that "the whole is greater than the sum of its parts," paving the way for more capable and cost-effective autonomous robots.

B

Let me go through the details or you can say my plans and then coming to the budget.

So usually in order to move a rover like the upcoming Musafir-3.0 you need to consider some few facts in my opinin:

1) How can we efficiently move the rover?
2) How can we let the rover sense its directions and avoid obstacles?
3) How can a rover recover from an obstacle he just faced and can't overcome?
4) How can we efficiently feed the rover the field datas of its surroundings?
5) Also it's a rover specifically designed for Mars so sensor datas and controlling distance should also be kept in mind.
6) And yes! The more efficiently we can produce our project with the lowest budget possible, the more recognizable and scalable it becomes.

For the perception layer or which we can call the the brain part of a rover:

We can set a stereo RGB-D camera, which will control the forward images or image collections through RGB-depth values. The device used for such cases are called depth sensing camera. So the advantage of using this camera is it has an IR camera that is responsible for converting the images into RGB values and extra 2 other cameras. SO even if your RGB camera gets shut down you can still use the other cameras for data collection while moving in mars. This is also actually one of the primary parts for ArUco.

So ArUco is used for square fiducial markers(Visual tags actually) in computer vision. They run on patterns. So each unique pattern has a unique ID number through which they can detect obstacles or use RGB values to analyse images. These marks can be easily detected using a camera and openCV ArUco module. Kind of like a QR code scanning.

Then I'll use IMU on the rover which is a sensor device that measures linear acceleration and angular velocity sometimes also magnetic fields. For a rover that is to perform on mars we have to think about its smooth movements and other stuffs such as turning or rotating. For example: For small bots like arduino controlled soccerbots or others; you'll see in order to turn or move they depends on only 3 axises: x y z . But in order to move in every direction also in between the x y z axises you need IMU to bend and smoothen your rover's controls.

Then I'll use LIDAR. In my opinion LIDAR can be used to find the inner gems or substances of the surface. Its actual work is to emit laser light pulses, which bounce off surfaces and return to the sensor. But what if we can use those pulses to get what is deeper inside the surface of mars in order to find traces of water or food resources ?! In that case we can use machine learning models by implementing various algorithms like regression, classifications etc in order to find the accuracy of those findings for future usage.

And yes use of AI for better performance:

We can use AI for object detection (YOLOv8/YOLO-Nano or MobileNet SSD) running on an edge GPU for the Object mission. Also we can use AI in ArUco based camera sensors where the model will be trained continuously through every datas it receives so that we don't have to manually declare them later.

And yes GPS is the core part of our rover. Without it we won't able to control the bot from our stations which will result in failure.

Now How the sensors & stack will handle the 3 mission types:

(i) Waypoint

We can Build a local SLAM map and maintain a continuous pose estimate. The mission gives global coordinates relative to base and then convert the mission coordinates to the local map frame using an initial alignment. Path planner uses SLAM map + LiDAR/depth for obstacle avoidance; arrival is reported when pose(rover's) error will be lesser than threshold.

(ii) ArUco tag scanning

As I said before we can use ArUco to use RGB stream + depth to detect and localize ArUco tags robustly. Depth gives distance to tag, enabling correct orientation for scanning,detection, ping base station with tag ID + pose + snapshot.

(iii) Object detection

We may have to Run a lightweight detector (YOLOv8-Nano, MobileNet-SSD or Tiny-YOLO converted to TensorRT) on Jetson as AI. Use RGB+D for better robustness.

Let's talk about the budget I come up with:

For the autonomous navigation system of **Musafir 3.0**, the focus is on balancing performance and affordability.

1) If we consider that the system is designed around the **NVIDIA Jetson Orin Nano Developer Kit** it may cost about ***36000tk*** which serves as the main compute unit capable of handling AI-based vision, SLAM, and ROS2 operations efficiently.

2) For environmental perception, the **Intel RealSense D435i** provides both RGB and depth data along with an internal IMU that may cost about ***30k tk***
3) For LIDAR(2D) we may use **RPLIDAR A3** offers 2D LiDAR for better mapping facility that may cost ***78k tk***.
4) To improve orientation and motion tracking, the **Bosch BNO055 IMU** is gonna be very helpful for its fast and efficient responses that may cost about ***4.8k tk***
5) **wheel encoders** may cost upto ***6k tk*** that will supply precise odometry data that can be fused with visual and LiDAR inputs for robust SLAM.
6) The **motor control system** may cost about ***1 lakh tk*** and if for a **custom aluminum chassis** may take ***180k tk*** but it will ensure stable and reliable movement across different terrains.
7) Power is supplied through a **Li-ion battery system** that may cost ***50k*** tk for its high performance and long-range communications can be handled by **Wi-Fi modules** that may cost upto ***20-30k tk.***
8) Additional storage and integration components may add around ***24,000tk.***

Overall, the estimated cost for the complete autonomous navigation setup stands at approximately ***5.5 lakh tk.***

So I think we may be able to get this amount of money depending on the previous missions of our rover's.

<u>C</u>

For me, LIDAR and Depth cameras are kind of similar in their nature but yes they do have some differences. And those may be efficient for the rover if being used properly.

For Musafir 3.0's upcoming missions (Waypoint, ArUco, Object Detection) and a tight budget, an RGB-D camera wins on practicality and cost. A RealSense gives **RGB + depth + built-in IMU** in one compact inexpensive package such as for D435i it may coust around ***30k tk***. So you keep depth sensing *and* the color stream needed for ArUco and object recognition without buying a separate camera. Depth-on-device reduces host CPU load and the extra RGB channel materially improves object detection and tag localization and helps distance measurements also.

Now Why not LiDAR ?

LiDAR excels at long-range,independent geometry and mapping but it's significantly more expensive, heavier, and *still requires* a separate RGB camera for ArUco/object tasks raising overall cost and integration complexity. With limited funds, LiDAR forces tradeoffs (you'd lose good visual perception or exceed budget).

But if you can manage to buy it you can use its efficiency to the highest for example you can use LIDAR to know about the ground surfaces of Mars for water or food resources for future.

(Just my opinions being shared : ")   )