ANALOG DEVICES

Industrial Mobile Robotics

Revolutionizing the Factory of the Future

VISIT ANALOG.COM/MOBILE-ROBOTICS

FOREWORD

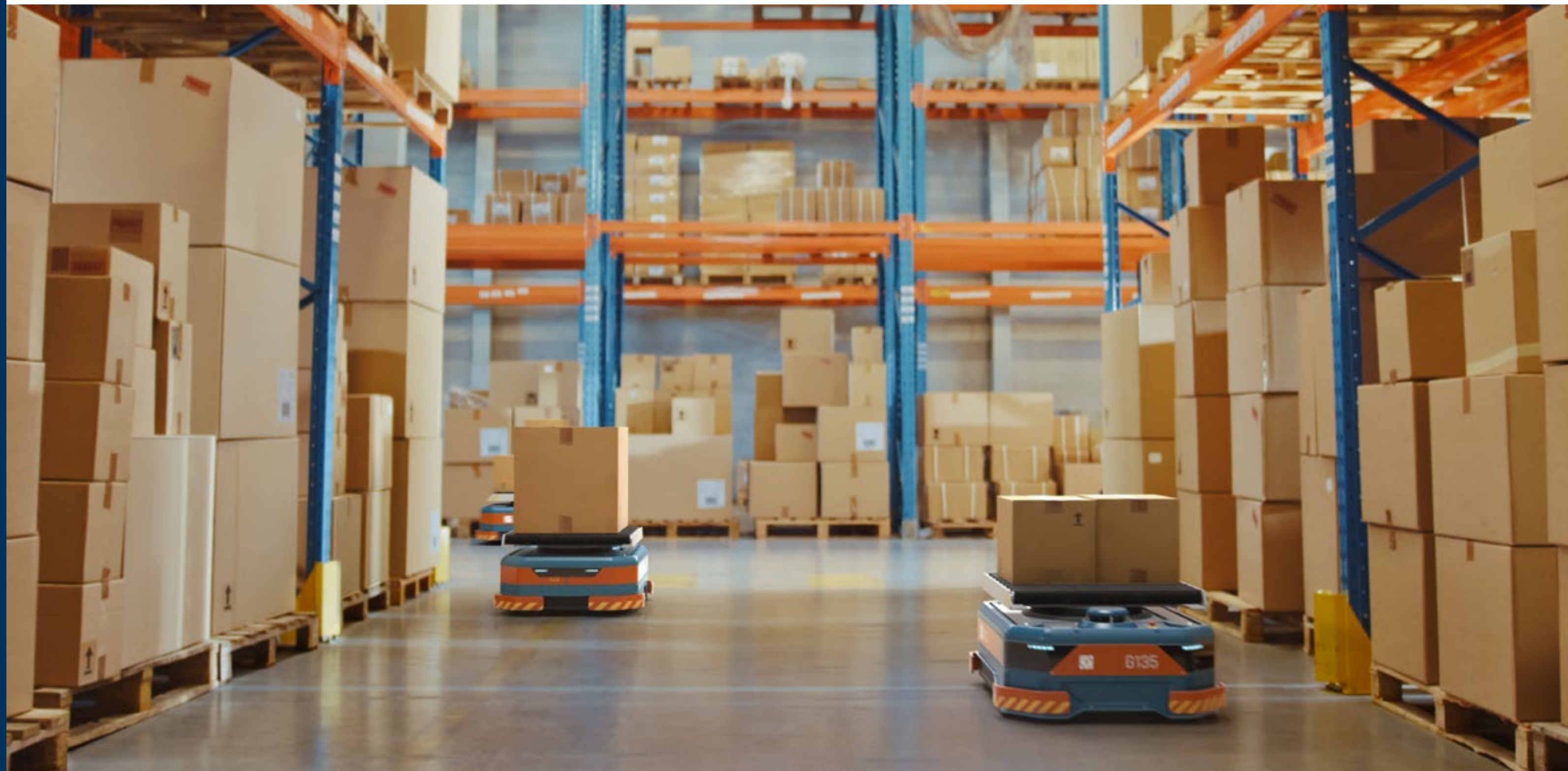
BY SHANE O'MEARA,
SENIOR APPLICATIONS MANAGER
SMART MACHINES & ROBOTICS,
INDUSTRIAL AUTOMATION,
ANALOG DEVICES INC.

Industrial robots have evolved from simple repetitive task performers to sophisticated, intelligent machines capable of autonomously learning, adapting, and complex decision-making. In today's manufacturing landscape, where increased automation is driving improvements in operational efficiencies within factories, warehouses and beyond, industrial mobile robots are required to adapt to different workflows and accommodate factory process changes more frequently.

Autonomous mobile robots (AMRs) and automated guided vehicles (AGVs) serve as intermediaries between humans and machines, linking disparate parts of the production process. With the help of advanced sensor technology and sophisticated software control, mobile robots are becoming more intelligent, safer, more versatile in navigating dynamically changing environments, and longer lasting thanks to efficient battery management systems.

This eBook aims to provide a holistic overview of the design challenges and technology innovations that are driving the evolution of industrial mobile robots – from understanding the right robotic fit for your industrial automation needs, to the exploration of critical topics such as localization, navigation, safety and motion control. Experts draw on ADI's rich history in industrial automation as well as their comprehensive system level knowledge to explore how key enabling technologies are addressing the challenges faced on the path to more efficient and flexible manufacturing capabilities.

CONTENTS



INDUSTRIAL MOBILE ROBOTICS – REVOLUTIONIZING THE FACTORY OF THE FUTURE TODAY

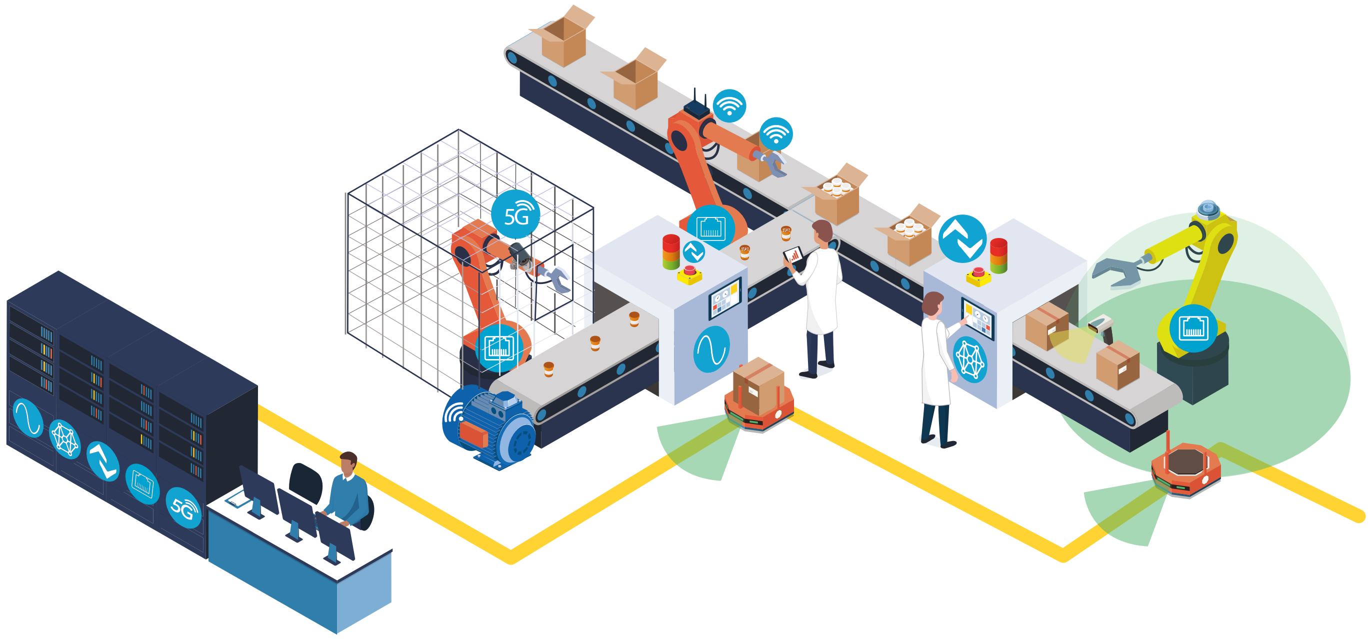
The automation of tasks is becoming commonplace in all facets of our lives. Who hasn't seen their neighbor out mowing their lawn lately but instead has seen a robotic lawnmower happily moving around the garden keeping the grass at bay. This is just one example of where automation in the form of a mobile robot is freeing humans from repetitive tasks. As much as I love not having to mow my lawn, the real impact of mobile robotics and the automation of tasks is most evident in factories and warehouses. Where constant movement of materials and assets within a manufacturing facility is required, here you see the power of what mobile robots can do.

Autonomous Mobile Robots (AMRs) or Automated Guided Vehicles (AGVs) allow disparate parts of a production process to be connected, they can be the go-betweens for humans or other machines facilitating the handoff from one station to another.

You might wonder "why are mobile robots making more of an impact today?" – the answer is simple; firstly, they are more **perceptive**. Data-driven insights are making a difference in today's automation revolution. In a true fusion of sensors, data from 3D time of flight cameras, Inertial Measurement Units (IMUs), LiDAR, and many more systems are collected and processed to create a real-time awareness of the robots' surroundings, seamlessly mapping the environment, and the robot's location within the factory floor.

> The real impact of mobile robotics and the automation of tasks is most evident in factories and warehouses.

Figure 1: *Industrial Production Line*



> Increased intelligence at the edge, improvements in battery management, and seamlessly connected factories are all supporting the deployment of mobile robotics.

Secondly, they are **efficient**. With advancement in real-time perception comes more freedom, and the ability to move around a dynamically changing environment. Warehouse and factory automation is relying on mobile robotics to improve efficiency, manage inventory, and speed up operations. In large warehouses, the management and movement of inventory is streamlined with mobile robots that can scan, locate, and move equipment and products, to ensure a 24/7 level of operation to meet demands for fulfilment, in the case of retail and online shopping. Another key factor is that fully automated factories can operate at lower temperatures and light conditions where humans normally would not work but robots "don't care".

With predictable movement comes the added benefit of safety. Having mobile robots transferring materials not only brings efficiency but also removes risks inherently involved in certain tasks. A task performed by a forklift controlled by a human has the potential for injury to the driver, to others in the location, and can be unpredictable in its operation. A mobile robot will consistently choose the most efficient way of getting from point A to point B, it will move slowly to its required location and, in the case, where accidents happen, no human is there to be injured. Some tasks may involve hazardous materials, or interaction with dangerous equipment, these scenarios pose no risk to a mobile robot tasked with transporting that material or getting close to a large piece of moving equipment, unlike a human operator tasked with the same job.

Thirdly, today's robots are more **flexible**, they can be programmed for new tasks as the needs arise in processes. With the quest for batch-size-of-one production facilities requiring customized processes, mobile robots that can adapt are vital. The Robot Operating System (ROS) which is a powerful developer tool for driver development has enabled many to begin their robotic journeys. Designing algorithms to bring human-like behavior to a mobile robot and the tasks that it must carry out. It's a fascinating time to be working in the field of robotics with more standardization, more collaboration, and the digital transformation happening in the industrial world – making mobile and mounted robots capable of understanding the environment, interacting, and communicating like never before.

Intelligence at the edge, improvements in battery management, and seamlessly connected factories are all supporting the deployment of mobile robotics. While some tasks are easier than others to automate, when it comes to the world of Industrial Mobile Robotics, the tasks are more complex, the environments more challenging and the intelligence onboard the robot will be key in its seamless integration into a highly automated manufacturing environment.

FINDING THE RIGHT FIT FOR YOUR INDUSTRIAL AUTOMATION NEED – AGVS OR AMRS

A mobile Automated Guided Vehicle (AGV) and an Autonomous Mobile Robot (AMR) share most of the same core fundamental attributes and will carry out similar automation tasks in an industrial application. The primary function for an AGV or an AMR is the transportation of materials from one location to another location in a factory or warehouse.

> The fundamental difference between an AGV and an AMR is in their navigation capability and ability to avoid obstacles.

The fundamental difference between an AGV and an AMR is in their navigation capability and ability to avoid obstacles. An AGV follows a predetermined track or route and will not deviate from this path as it travels from one location to another. If an AGV detects an obstacle blocking the path it will stop and remain in place until the obstacle is removed. An AMR also has the ability to detect an obstacle but has the intelligence to adapt its route to avoid the obstacle, calculate a new route and proceed to complete its goal without user intervention.

An AGV can use different sensing techniques to detect its route.

- **Magnetic tape** stuck onto the factory floor. A corresponding sensor under the AGV detects the location of tape and adjusts its position. Additional pieces of magnetic tape can be used to code locations.
- **Inductive wire** embedded in the floor. Again, a sensor detects the wire and adjusts the AGV position accordingly.
- **Visual tracking**, colored tape or markers such as AprilTags are placed on the ground and detected with RGB cameras to map route and determine location.
- **Laser guided** by a 360° laser mounted on the AGV and several reflectors installed in the facility. The AGV measures the distance and angle to the reflectors and triangulates its position.

The main advantage of AGVs is that they follow their predetermined route precisely and consistently hence making them ideal for high volume, repetitive tasks. Their main disadvantages are that they require infrastructural changes within the facility they intend to operate in and significant effort to setup and maintain. Tape, wire, markers or reflectors must be installed on the floor or walls and then maintained. Visual markers can be impacted by dirt or magnetic tape may become dislodged. Inductive wire is extremely robust, but any adjustments have a significant impact on day-to-day operation as the floor must be redone.

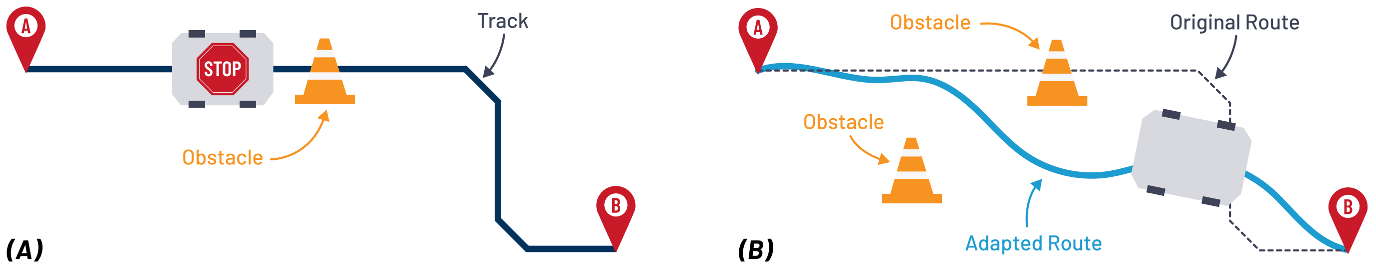


Figure 1:
(A) *AGV Operation with Obstacles*
(B) *AMR Operation with Obstacles*

Additionally, as mentioned previously, AGVs lack the ability to avoid obstacles, so they stop operating if an obstacle is detected.

AMRs use Simultaneous Localization and Mapping (SLAM) to navigate the factory floor. Depth sensors, commonly LiDAR scanners, mounted on the AMR are used to map the factory floor. The AMR is first driven around the facility and scans are accumulated to generate a complete map which is stored on the AMR and in fleet management control software.

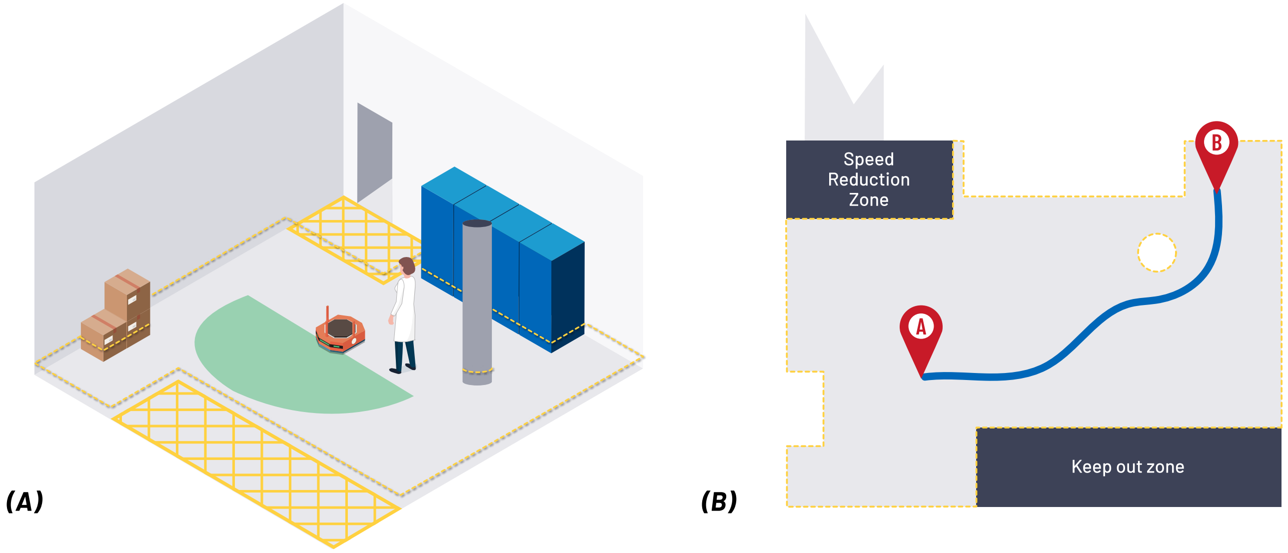


Figure 2:
(A) *AMR Mapping*
(B) *Generated Map*

The generated map can be enhanced with additional information such as keep out zones, speed reduction areas and docking station locations. Goals can be placed on the map as x, y coordinates or dropped pins for the AMR to navigate between. During operation the most recent scan from the LiDAR scanners is compared against the stored map and the current AMR position and orientation are calculated. The AMR then uses the built-in navigation system to determine the optimized route to its goal considering the stored map and any obstacles it encounters on the path.

Disadvantages of AMRs compared to AGVs is that they have a higher initial cost and are less predictable than an AGV which takes a defined time to reach its goal. The obvious advantage of AMRs is that there is no requirement for infrastructural changes to enable their operation as an AMR has the sensing capability and intelligence to localize and navigate autonomously without markers. New tasks and goals can be updated within the map or fleet management software quickly and efficiently. Facility expansions and automation upgrades can be easily accommodated with the generation of new maps or extension of existing maps.

In the dynamic environments of the factory of the future, the ease of use, flexibility, and scalability of an AMR gives it a clear advantage compared to AGVs. AMRs generally have enhanced sensing capabilities, with longer range LiDAR, 3D depth sensing, radar and RGB vision technologies. These sensing modalities combined with superior compute power and artificial intelligence open up additional possibilities for advanced features and improved human robot interaction.

LOCALIZATION: THE KEY TO TRULY AUTONOMOUS MOBILE ROBOTS

We know from the previous chapter that superior sensing modalities and compute power give Autonomous Mobile Robots (AMRs) an advantage over Automated Guided Vehicles (AGVs) when navigating dynamic environments. In this chapter we are going to explore why that is the case.

> For mobile robots, the ability to map their surroundings and identify their position relative to that map are key.

Localization is the process of determining where a robot is located within its environment. For mobile robots, the ability to map its surroundings and identify its position relative to that map are key. With greater localization awareness, tasks can be performed faster and more efficiently, as the majority of a mobile robot's tasks involve moving from one location to another. It is this freedom of movement that gives AMRs independence within a factory, but how does it work?

INTRODUCTION TO INERTIAL MEASUREMENT UNITS (IMUs)

Inertial Measurement Units (IMUs) provide crucial motion data for precise robot positioning. Integrated **accelerometers** measure acceleration with respect to the earth's gravitational field, **gyroscopes** measure the rate of rotation providing angular velocity, and **magnetometers** support accurate orientation estimation in challenging environments. By integrating all three of these advanced sensing technologies, IMUs enable robots to precisely determine their orientation, position, and movement.

Figure 1: *AMRs in an industrial factory setting*

Let's consider the challenges for localization and how IMUs overcome these.

Dead Reckoning: A navigation technique to estimate current position based on a previously known position. By constantly providing data on position, orientation, and speed over elapsed time, IMUs enable precise estimation, contributing to reliable navigation for AMRs.

Robustness: Environmental factors can have a significant impact on sensor performance. LiDAR sensors, for instance, may exhibit sensitivity to ambient light, dust, fog, and rain, resulting in decreased sensor data quality and potential disruptions in performance. Other sensor modalities, such as cameras, may encounter challenges from reflective surfaces and dynamic obstacles like other AMRs or workers. In contrast, IMUs demonstrate robustness across diverse conditions, including environments with electromagnetic interference, enabling them to operate effectively both indoors and outdoors. This adaptability makes IMUs an optimal choice for mobile robots, ensuring consistent performance in the face of environmental complexities.

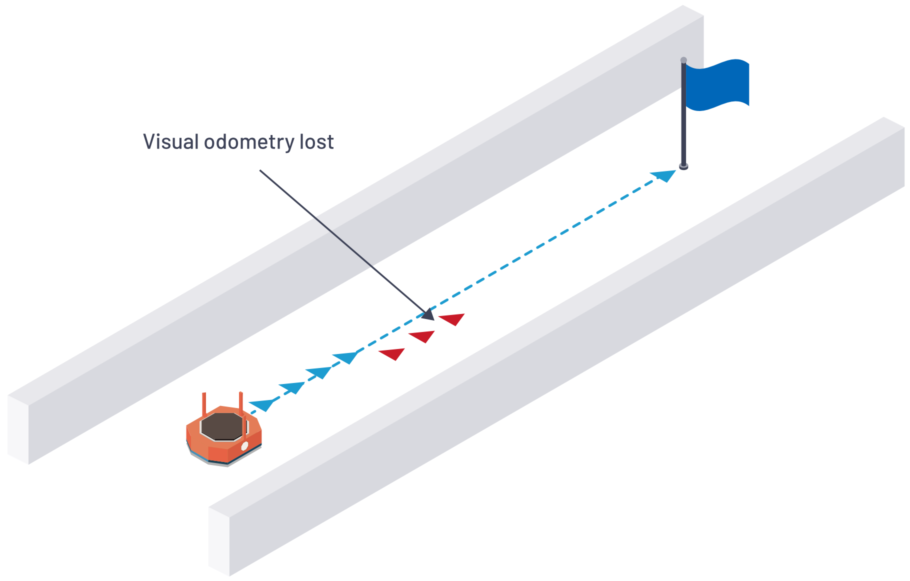
Enhanced Reliability: IMUs stand out by providing high-fidelity positional output of up to 4 kHz raw data. Other perception sensors are typically limited to update rates of ~10 Hz to 30 Hz. This increased update rate enhances reliability of IMU performance, especially in dynamic environments, enabling AMRs to estimate their position quickly and accurately in the short time between other measurements.

IMUs VERSUS VISUAL ODOMETRY

You might be wondering, with the advancement in vision systems, why is the IMU still playing such a pivotal role in mobile robotics? Here's why:

SLAM (Simultaneous Localization And Mapping) algorithms match observed sensor data with stored data to localize within the map. But what happens when observed sensor data is limited, for example in a long corridor with straight walls of uniform color, texture, or reflectivity? SLAM algorithms can struggle to localize precisely in such environments, and the AMR is likely to lose its position quickly due to a lack of distinctive features.

Figure 2: *AMR Mapping*

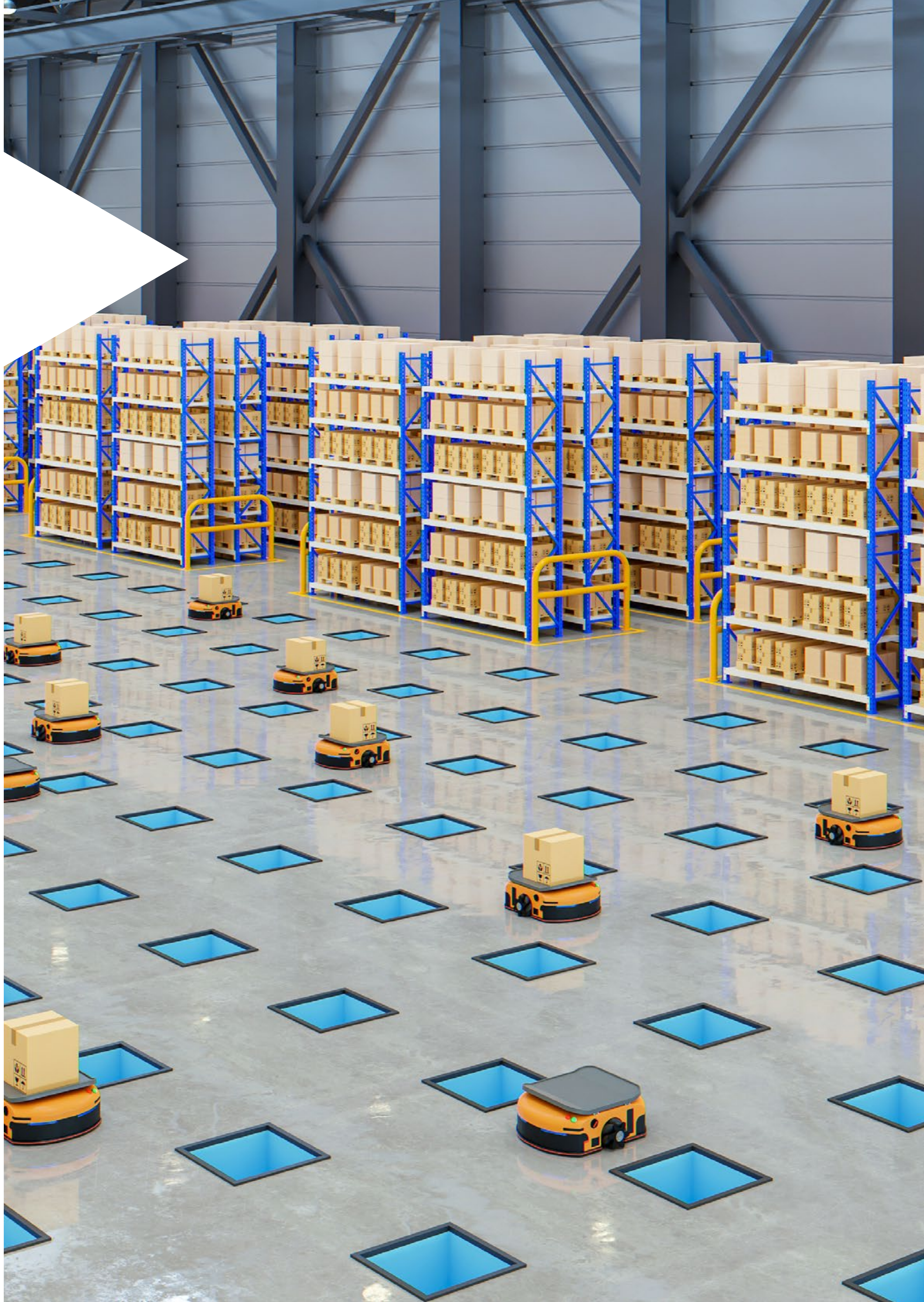


IMUs act as a valuable guidance system by providing heading and orientation information in feature-sparse environments such as corridors. They provide high short-term accuracy and immediate measurements between vision sensor measurements. IMUs have lower computational needs than visual odometry, enhancing redundancy and further endorsing them for AMR operations.

IMUs: PART OF A HOLISTIC AMR DESIGN

While IMUs offer many benefits over other sensors, they can also be prone to drift. In situations where the environment is constantly changing, it may be advantageous for AMR operations to rely on multiple sensor inputs. This allows each sensor to overcome the limitations of the others for greater success. The next blog of the series will explore how this sensor fusion works and the combined benefit it brings to robotics.

Factories of the future may be purpose built and optimized for AMRs to operate in, but adapting these robots to existing warehouses and factories presents challenges.



STOP, LOOK, LISTEN – HOW DO AUTONOMOUS MOBILE ROBOTS NAVIGATE THEIR ENVIRONMENTS?

We discussed the important role that IMUs play in localization for autonomous mobile robots (AMRs) in the previous chapter. Here we will elaborate on how navigation relies on a fusion of sensor technologies working together to allow AMRs true freedom within dynamically changing environments.

> AMRs use Simultaneous Localization and Mapping (SLAM) techniques to navigate

So how do mobile robots learn to get around? As kids we are all thought to "stop, look and listen" before crossing a road, but does this same concept apply to robots. As humans we rely on our eyes and ears to help us "navigate" our environment, robots on the other hand use sensors to provide an awareness of their surroundings.

AMRs use Simultaneous Localization and Mapping (SLAM) techniques to navigate. The process involves the AMR being driven around the facility and scanning its environment. These scans are combined and generate a complete map of the area. AMRs utilize an array of sensors and algorithms for localization and navigation. Sensor technology such as industrial vision time-of-flight cameras, radar and LiDAR are the "eyes" of an AMR, combined with data from IMUs and wheel odometry (position encoders).

However, no single sensor is perfect. The true power lies in the diverse sensor types working together to produce effortless navigation in dynamically changing environments.

Each sensor has strengths and weaknesses that are balanced out by having more than one sensor type being relied on for navigation purposes. Let's consider how multiple sensors can enhance the overall AMR performance.

ENVIRONMENTAL FACTORS WHILE NAVIGATING

LiDAR sensors can be sensitive to various environmental factors, such as ambient light, dust, fog, and rain. These factors can degrade the quality of the sensor data and, in turn, affect the performance of the SLAM algorithm. Similarly, other sensor modalities can be affected by reflective surfaces, dynamic moving objects (other AMRs or workers) thus further confusing SLAM. The table below summarizes how environment affects different sensors modalities.

Sensor Modality	Affected by Poor Lighting	Affected by Dynamic Movers	Affected by Reflective Surfaces	Reliant on Rich Scene Geometry
Standard RGB Camera	*Yes*	*Yes*	*No*	*No*
Time-of-Flight (ToF)	*No*	*Yes*	*Yes*	*Yes*
LiDAR	*No*	*Yes*	*Yes*	*Yes*
Radar	*No*	*Yes*	*Yes*	*Yes*
Wheel Odometry	*No*	*No*	*No*	*No*
IMU	*No*	*No*	*No*	*No*

Table 1: *Comparison table of sensor modalities*

While IMUs and wheel odometry are not affected by visual elements within the working environment, the use of this sensor data in conjunction with visual data means the AMR can operate better in any scenario encountered. Let's consider the challenge of navigating on a sloping floor surface.

NAVIGATING ON A SLOPE

While maneuvering on a slope, traditional SLAM algorithms encounter challenges when relying on LiDAR, as the 2D point data does not show gradient information. Consequently, slopes are misconstrued as walls or obstacles, leading to higher cost maps. As a result, conventional SLAM approaches with 2D systems become ineffective on slopes. IMUs help to solve this challenge by extracting gradient information to effectively negotiate navigating on a slope.

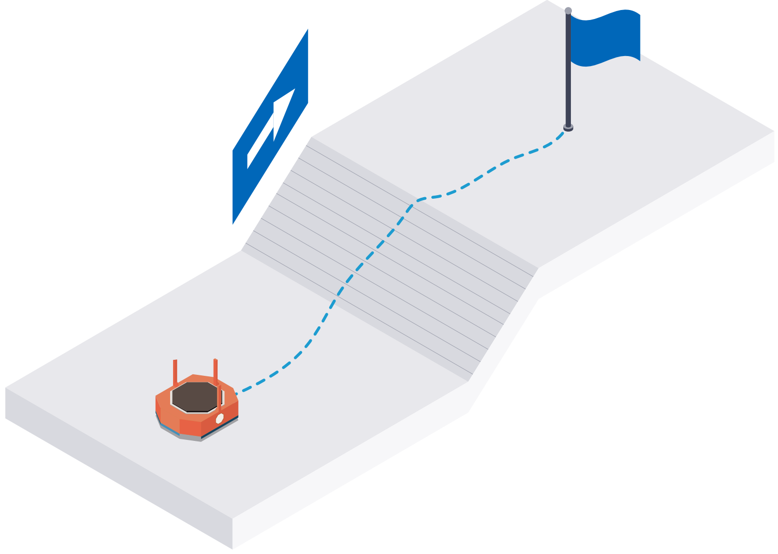


Figure 1: *AMR Mapping*

HOW DOES THE SENSOR DATA GET COMBINED?

In a typical ROS (Robot Operating System), vision sensors along with IMU and wheel odometry are combined through a process called sensor fusion. A widely used open source ROS package is robot_locatlizaton1 which utilizes EKF (Extended Kalman Filtering) algorithms at its core. By fusing data from diverse sensors such as LiDAR, cameras, IMUs, and wheel encoders, EKF helps in better estimating and understanding of the robot's state and its environment. Through recursive estimation, EKF refines the robot's position, orientation, and velocity while simultaneously creating and updating a comprehensive map of the surroundings. This fusion of sensor data enables mobile robots to overcome individual sensor limitations and navigate complex terrains with greater precision and reliability. By leveraging techniques like EKF help in collective insights of sensors, deriving meaningful sensor fusion of various sensor modalities allowing mobile robots to effectively perceive and interact with their environment and help navigate AMRs autonomously.

A future chapter will cover the Robot Operating System in more detail. However, the focus of this chapter is to leave you confident that sensor fusion offers increased reliability, increases the quality of data, while providing greater safety for objects and people within the environment as AMRs aren't relying on a single means to navigate.

REFERENCE / RESOURCES:

[1] https://docs.ros.org/en/melodic/api/robot_localization/html/index.html

For the third year in a row, annual robot installations exceeded the

500k unit mark.

- World Robotics Industrial Robots 2024 Report published by the International Federation of Robotics

STAYING CONNECTED WHEN ALWAYS ON THE MOVE – THE COMMUNICATION BACKBONE OF MOBILE ROBOTS

We previously discussed how mobile robots navigate their environments through a variety of sensor types to collect data about the real world. However, we didn't explore how this data is communicated within the robot to translate perception into actions, such as movement.

Mobile robots consist of various technologies that must communicate with each other quickly and reliably to transmit critical messages for navigation and performing tasks, whether it's an Autonomous Mobile Robot (AMR) or an Automated Guided Vehicle (AGV). For a refresher on the differences between these types of robots, you can refer to the earlier chapter on AGVs or AMRs. Let's consider the architecture of an AMR as shown:

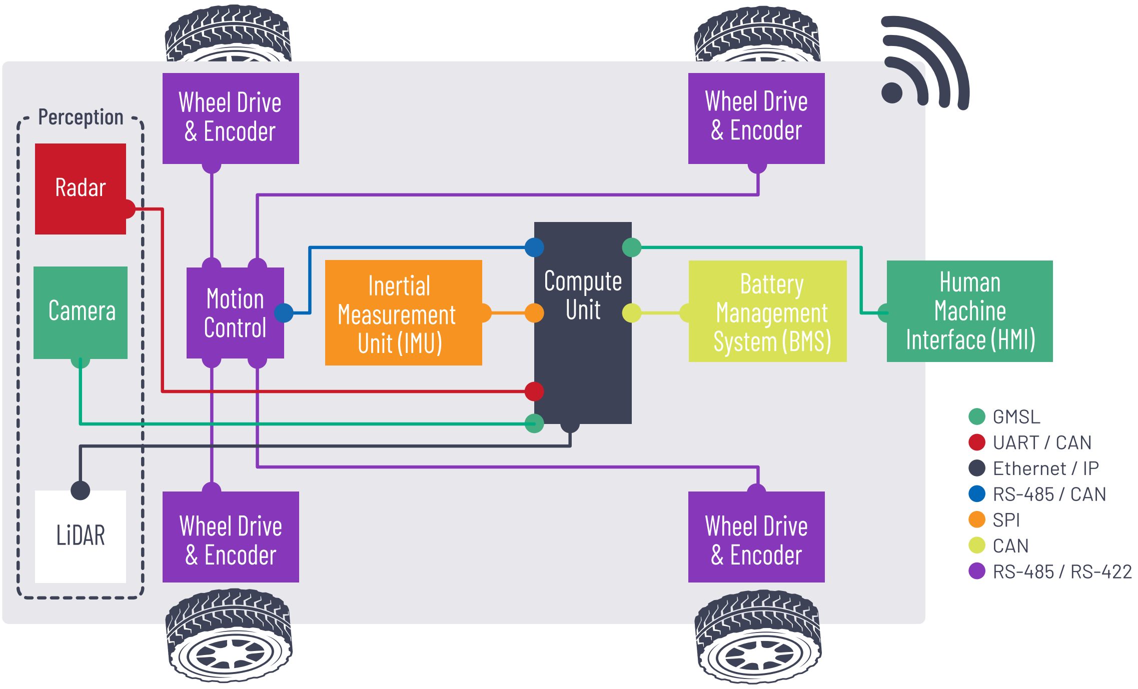


Figure 1: *AMR Communications Overview*

There are several components that make up any mobile robot (such as wheel drive & encoder systems, vision inputs, inertial measurement unit (IMU) data, and battery management systems), and all of them need to communicate, usually with a main controller or main compute unit or sometimes to decentralized units that control specific functions of the robot, which can be done to reduce the overhead on a main controller and also aid in time critical applications such as perception of its environment and actuator control. There are many communication methods that live within the operation of a typical mobile robot, and each type of protocol has their pros and cons for use. In the above example there are potentially seven different communication methods employed within the one mobile robot: GMSL, UART, CAN, Ethernet, RS-485, SPI, RS-422.

> Operations in mobile robots typically demand near real-time speeds to function effectively.

While this chapter focuses on wired communication protocols, it is important to note that mobile robots typically require wireless communication as well. Wireless communication is essential for enabling mobile robots to interact with a base station and collaborate with other robots, ensuring seamless coordination and operation in dynamic environments.

Here is a quick comparison of a selection of technologies comparing their speed and latency.

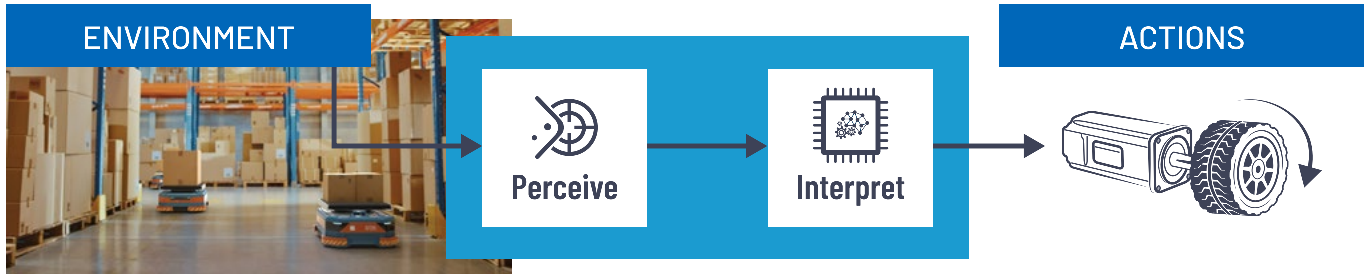
Protocol	Type	Speed	Latency
PROFIBUS	Fieldbus	Up to 12 Mbps	High
PROFINET	Industrial Ethernet	Up to 100 Mbps	Low
EtherCAT	Industrial Ethernet	Up to 100 Mbps	Very Low
Ethernet/IP	Industrial Ethernet	Up to 100 Mbps	Low
Gigabit Multimedia Serial Link (GMSL)	Serial	Up to 6 Gbps	Very Low

Table 1: *Protocol Speed/Latency Comparison Table*

As it can be seen in *Table 1*, the parameters for the highlighted technologies vary in speed and latency and the appropriate technology needs to be chosen according to the need and the design itself and will most likely include a combination of different technologies. Operations in mobile robots typically demand near real-time speeds to function effectively. This is crucial for tasks such as obstacle avoidance, navigation, and interaction with dynamic environments, where even slight delays can impact performance and safety. The key parameters that need to be taken into consideration for communication are performance, reliability, and scalability.

An AMR needs to be able to navigate while perceiving its surroundings to execute tasks in an efficient way, and a simple flow diagram can describe how it acts:

Figure 2: *AMR Navigation Functions*



Both the perception and the action parts play important roles, the environment needs to be perceived in order for actions to be taken and this data is usually acquired with RGB cameras, depth cameras, LiDAR sensors and radar or a combination but transferring all this data to a processing unit needs a robust link with enough bandwidth and in the case of industrial robots, reliability against interferences. That critical work can be executed by protocols such as GMSL.

GIGABIT MULTIMEDIA SERIAL LINK (GMSL)

There is a new protocol entering the mobile robotics scene, GMSL. The protocol can transfer up to 6 Gbps of advanced driver assistance systems (ADAS) sensor data over a coax cable while simultaneously transferring power and control data over a reverse channel. It is a highly configurable serializer deserializer (SERDES) interconnect solution which supports sensor data aggregation (Video, LiDAR, Radar, etc.), video splitting, low latency and low bit error, and Power over Coax (PoC)

The topology for a GMSL application consists of the sensor, a serializer, a cable, and a deserializer on the system on chip (SoC) side.

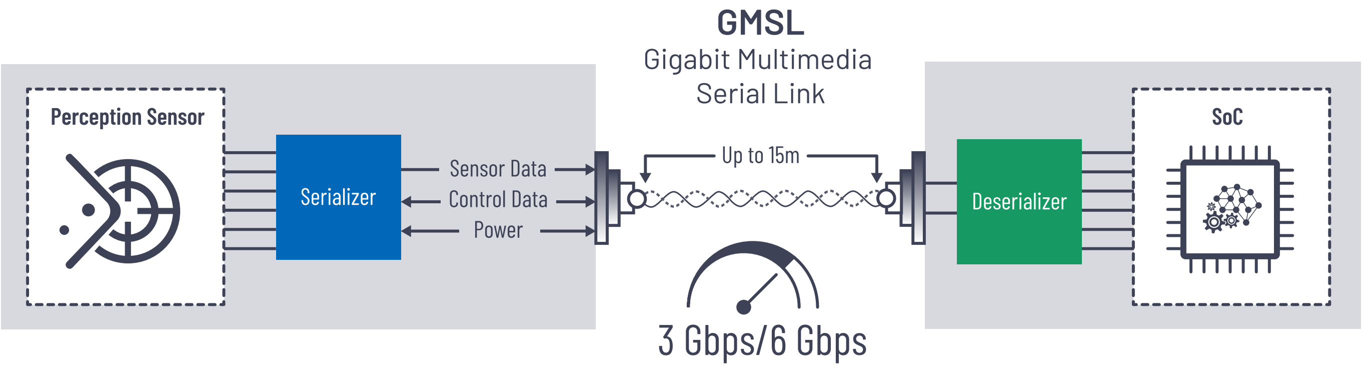


Figure 3: *GMSL Topology*

This simplifies the mobile robot design and makes it more robust since GMSL was designed with transferring this type of data and was optimized to ensure high bandwidth, low latency transmission of data.

The synergy between Industrial Ethernet, GMSL, and wireless communication technologies is driving the next generation of mobile robotics. These technologies provide the robust, high-speed, and flexible communication necessary for mobile robots to operate autonomously and efficiently in various environments. As innovations continue to emerge, the capabilities of mobile robots will expand, revolutionizing industries and transforming our daily lives.

EVER READY – HOW MOBILE ROBOTS ARE LASTING LONGER THROUGH EFFICIENT BATTERY MANAGEMENT

In the fast-paced world of automated warehouses and manufacturing facilities, every second counts. Autonomous mobile robots (AMRs) and Automated Guided Vehicles (AGVs) are crucial to this ecosystem, where even a brief downtime can cause significant disruptions. Ensuring the efficient and safe operation of these robots requires meticulous control over their components, especially their batteries.

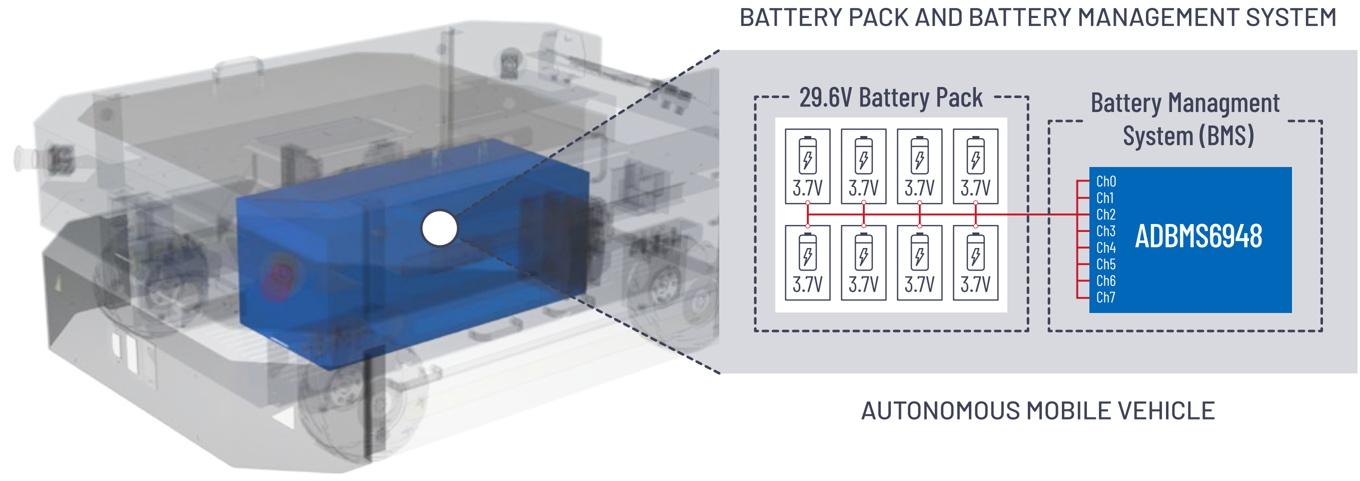


Figure 1: *Mobile Robot Battery Management System*

THE IMPORTANCE OF BATTERY MANAGEMENT SYSTEMS

A Battery Management System (BMS) is an electronic system designed to monitor and control various parameters of a battery pack or its individual cells. These systems are essential for maximizing the usable capacity of batteries while ensuring safe and reliable operation. An efficient BMS can provide accurate measurements of cell voltage, **state of charge** (SoC), **depth of discharge** (DoD), **state of health** (SoH), temperature, and current. These parameters are vital for optimizing the performance of mobile robots.

KEY METRICS IN BATTERY MANAGEMENT

- State of Charge (SoC): This parameter indicates the current charge level of the battery relative to its total capacity. It is expressed as a percentage, with 0 % being empty and 100 % being fully charged. Accurate SoC measurements are crucial for planning the operational cycles of mobile robots and preventing unexpected downtimes.
- State of Health (SoH): SoH represents the maximum capacity of the battery compared to its rated capacity. It helps in assessing the aging and overall health of the battery, which is vital for maintenance and longevity.
- Depth of Discharge (DoD): This is the opposite of SoC and indicates the percentage of the battery that has been discharged relative to its rated capacity. Managing DoD is crucial to prevent over-discharging, which can damage the battery and reduce its lifespan.

WHY THESE METRICS MATTER FOR AMRs AND AGVs

Different types of batteries, such as Lithium-Ion (Li-Ion) and lead-acid batteries, offer various benefits and challenges. Li-Ion batteries, for instance, have a higher energy density, are lighter, charge faster, and have a longer life cycle compared to lead-acid batteries. However, they are also more expensive and require precise monitoring to realize their full potential.

For example, a typical LiFePo4 battery pack used in an AMR has a specific voltage range that needs to be monitored accurately. Even a minor measurement error can significantly affect performance. If a LiFePo4 battery pack for a 24 V system with a capacity of 27.2 V has a measurement error, it can lead to underutilization and increased costs. Accurate SoC and DoD measurements are therefore critical for optimizing battery usage and extending its life.

In *Figure 2* we see a plot of the depth of discharge for two common types of batteries, a Li-Ion battery, and a Lead Acid battery. As it is possible to observe, the voltage variation for the Li-Ion is very small, while going from 10 % to 90 % of DoD, from that we can infer that a few mV can have a big impact on a precise measurement thus improving efficiency.

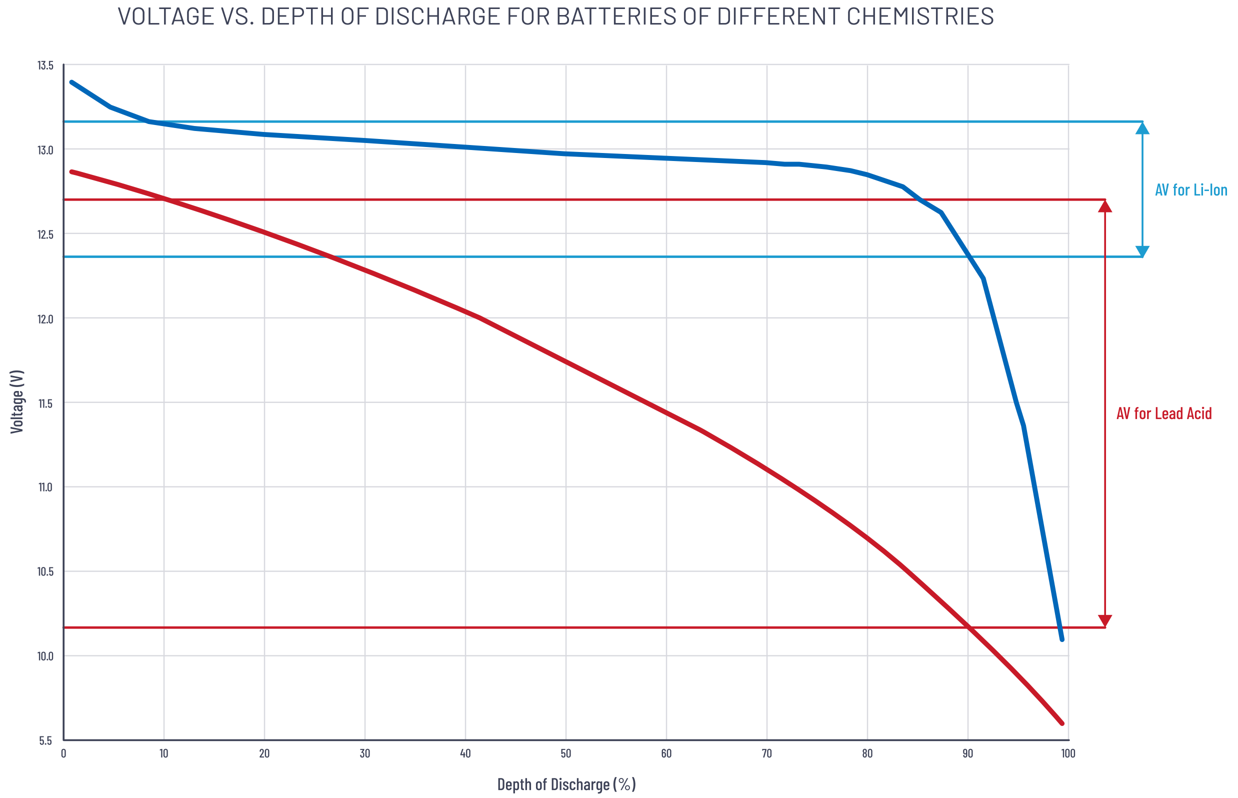


Figure 2: *Voltage vs. Depth of discharge for batteries of different chemistries*

> Efficient battery management is not only about improving performance but also about reducing waste and environmental impact.

Another important topic is the natural degradation of batteries, as we can see in *Figure 3*, the full range of operation of a battery will decrease within its lifetime so precise voltage measurements and SoC, DoD estimations become paramount.

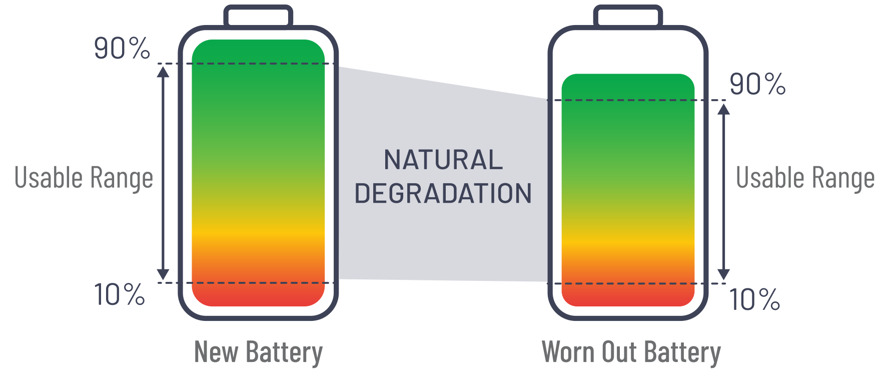


Figure 3: *Battery Range of Operation Natural Degradation*

REDUCING WASTE AND ENVIRONMENTAL IMPACT

Efficient battery management is not only about improving performance but also about reducing waste and environmental impact. Citing to the International Energy Agency report of 2023, "Batteries are essential for the clean energy transition". Proper management can extend battery lifespan, reducing the need for frequent replacements and minimizing environmental damage from battery disposal.

Figure 4 shows some of the most critical degradation factors for batteries, all of them can be controlled and/or monitored with the proper technologies.

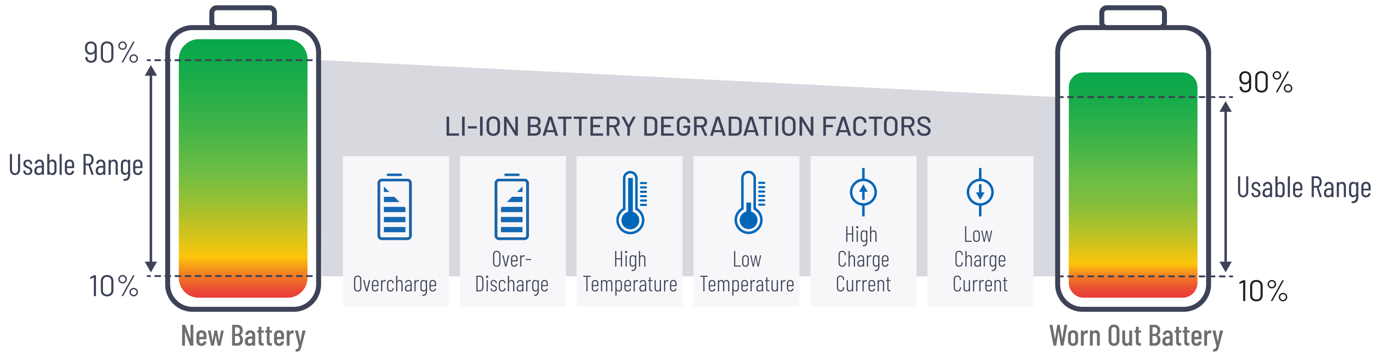


Figure 4: *Battery Degradation Factors*

ADI'S BMS SOLUTIONS: ENHANCING EFFICIENCY AND SAFETY

ADI's BMS solutions (ADBMS6948) offer advanced technologies that enhance the performance and safety of mobile robotics applications. These systems provide precise cell measurements, balanced charging, synchronous current and voltage measurements, and fast over-current detection to optimize battery performance, extend battery life, and enhance safety.

For instance, the ADBMS6948 offers several key features, including:

- Low total measurement error (TME) over the battery's lifetime
- Simultaneous and continuous cell voltage measurements
- Built-in isoSPI interface for robust communication
- Hot plug tolerance without external protection
- Passive cell balancing for uniform charge distribution
- Low power cell monitoring and low sleep mode supply current

These features ensure that the batteries are used efficiently and safely, reducing the risk of overcharging, over-discharging, and overheating.

CONCLUSION

In the evolving landscape of automated manufacturing and warehousing, the efficiency and reliability of mobile robots are paramount. ADI's Battery Management Solutions play a critical role in enhancing the performance, safety, and longevity of these robots by providing precise monitoring and control of battery parameters. By extending battery life and reducing waste, these solutions not only improve operational efficiency but also contribute to environmental sustainability. As automation continues to advance, the importance of robust and efficient battery management systems cannot be overstated. To learn more visit *analog.com/multicell-battery-stack-monitor*

WITHOUT INTELLIGENT MOTION CONTROL, MOBILE ROBOTS ARE GOING NOWHERE

As the ecosystem for autonomous mobile robots (AMRs) gets more and more complex, and as the design complexity of robotic systems also increases, efficient movement remains at the core of the architecture.

The communications methods discussed in a previous chapter outlined how mobile robots need to perceive their environment, communicate with several internal components, and make timely decisions. These factors combine to enable intelligent motion control as summarized in *Figure 1*. Smarter and more efficient motor controllers allow for less energy waste, which is a key factor since mobile robots are battery operated.

Smarter and more efficient motor controllers allow for less energy waste, which is a key factor since mobile robots are battery operated.

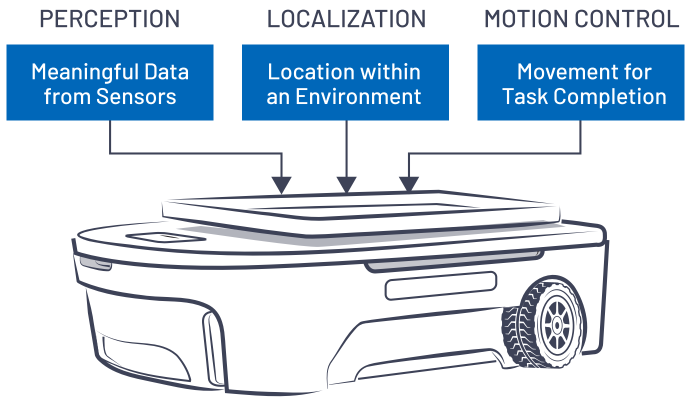


Figure 1: *Core blocks for mobile robot intelligent motion control*

CALCULATING ACTUATOR REQUIREMENTS

Mobile robots can serve various applications and move different types of loads. Therefore, they vary widely in size and shape, and so do their drive systems. Usually, the design of such systems requires a few steps to be taken. Let's walk through a simplified scenario to describe how the decision workflow could be done. Imagine you're designing a robot that needs to move a payload of $m_{payload}$ KGs with an AMR that weighs m_{AMR}.

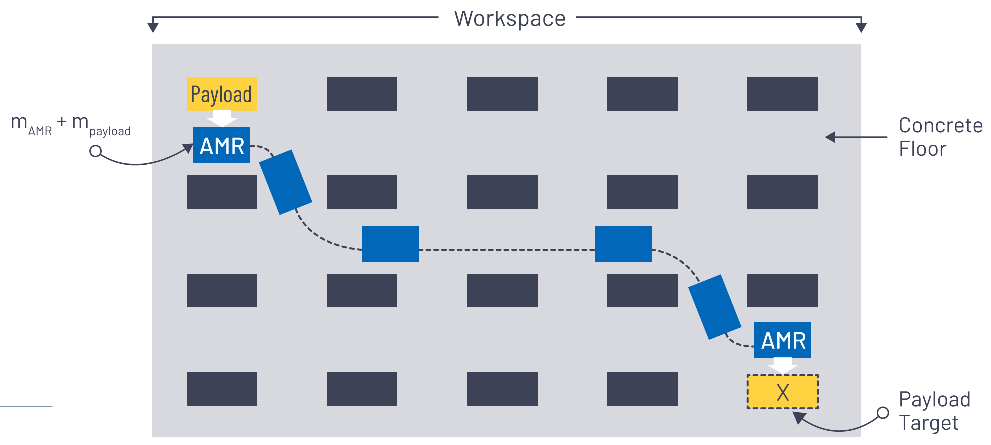


Figure 2: *Example scenario for AMR operation*

The first step is to understand the application and constraints of the project. Since the beginning of the design is mostly mechanical, we won't dive into the specifics here, but for the purpose of this example, the following needs to be considered:

- m_v: The expected vehicle mass.
- D_w: Wheel outer diameter.
- m_w: Wheel mass.
- Drive type: Differential or Omnidirectional.
- Number of wheels: 2 for differential, 4 for omnidirectional.
- Rolling friction between wheel and floor, usually hard to calculate but estimates can be made from available literature sources.
- v_{max} and v_{min}: Maximum and minimum speed.
- n_{system}: System Efficiency.
- SF: Safety Factor.

INERTIA CALCULATIONS

From these considerations, it is possible to calculate the motor specifications for this application, starting with the **inertia** for the vehicle.

$$J_{Total} = \left((m_V + m_L) + \frac{1}{2} \times m_W \times n_W\right) \times \left(\frac{D_1}{2}\right)^2$$

This inertial measurement is considering a drive straight out of the motor shaft, if we have a gearbox, where GR is the gear ratio. The inertia seen by the motor is called **reflected inertia** and can be calculated by:

$$J_M = J_{Total}\left(\frac{1}{GR^2}\right)$$

TORQUE CALCULATIONS

Next, we need to calculate the **acceleration torque**. To get to maximum speed, we need to accelerate a load as it is shown in *Figure 3*, which relies on the permissible inertia of the rotor.

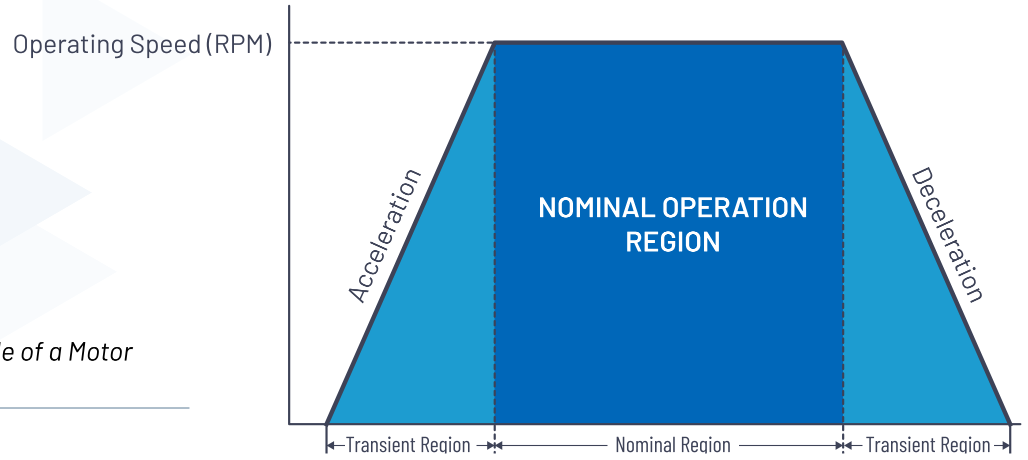


Figure 3: *Speed Profile of a Motor in an AMR application*

The **acceleration torque** can be calculated by:

$$T_a = J_M \times A$$

Where J_m is the reflected moment of inertia and A is the acceleration rate.

We then need to calculate the **load torque** (T_L), which can be calculated using:

$$T_L = F \times \frac{D_W}{2 \times \eta}$$

Then we can calculate the **total torque**.

$$T_{total} = (T_a + T_L) \times SF$$

MOTOR CONTROL

There are many other parameters to calculate for a mobile robot design, but the rest are outside the scope of this chapter. Next, we will consider how to control the motor. There are several methods, but one of the most efficient ways of controlling Brushless DC motors is through **Field Oriented Control (FOC)**.

FOC leverages real-world current and rotor position information to drive electric motors using orthogonal applied current. It generates a precise amount of torque by controlling the current needed to generate the target torque and phasing the magnetic field of magnets within the rotor. Thus, FOC can manage the torque of 3-phase motors with high accuracy and bandwidth.

FOC works by calculating a vector for the orthogonal current IQ using an accurate measurement of the position of the rotor. It then uses the current ID to control the magnetic flux inside the motor. Finally, we use two proportional and integral (PI) controllers, a torque controller which controls IQ and a magnetic flux controller which controls ID. A more detailed explanation can be found here.

CHOOSING THE BEST SOLUTION FOR YOUR AMR

A motor control solution needs to fit the requirements of the project—not only the motor's power requirements but also the interface, safety requirements, efficiency, and more. The challenges could be quickly reduced by choosing solutions that will offer the flexibility needed for the design.

> A motor control solution needs to fit the requirements of the project—not only the motor's power requirements but also the interface, safety requirements, efficiency, and more.

For example, ADI's TMCM-2611-AGV module is a dual-axis servo drive platform for 3-phase BLDC motors running with up to 14 A RMS at +48 V, which has been designed for motion control in mobile robots. A few of its benefits include:

- **Hardware based FOC control:** By integrating the control onto a hardware-based solution, overhead on external processors is reduced.
- **Flexible communications:** Many communication protocols are integrated, allowing for flexibility in the design. It supports RS-485, CAN and USB using the TMCL protocol.
- **Feedback sensors:** Supporting options such as incremental quadrature encoder and digital hall sensor.
- **Motor Brakes:** The importance of this feature will be outlined n our next blog addressing mobile robot safety standards.

To learn more about Analog Devices solutions for motion control, visit *analog.com/intelligent-motion*

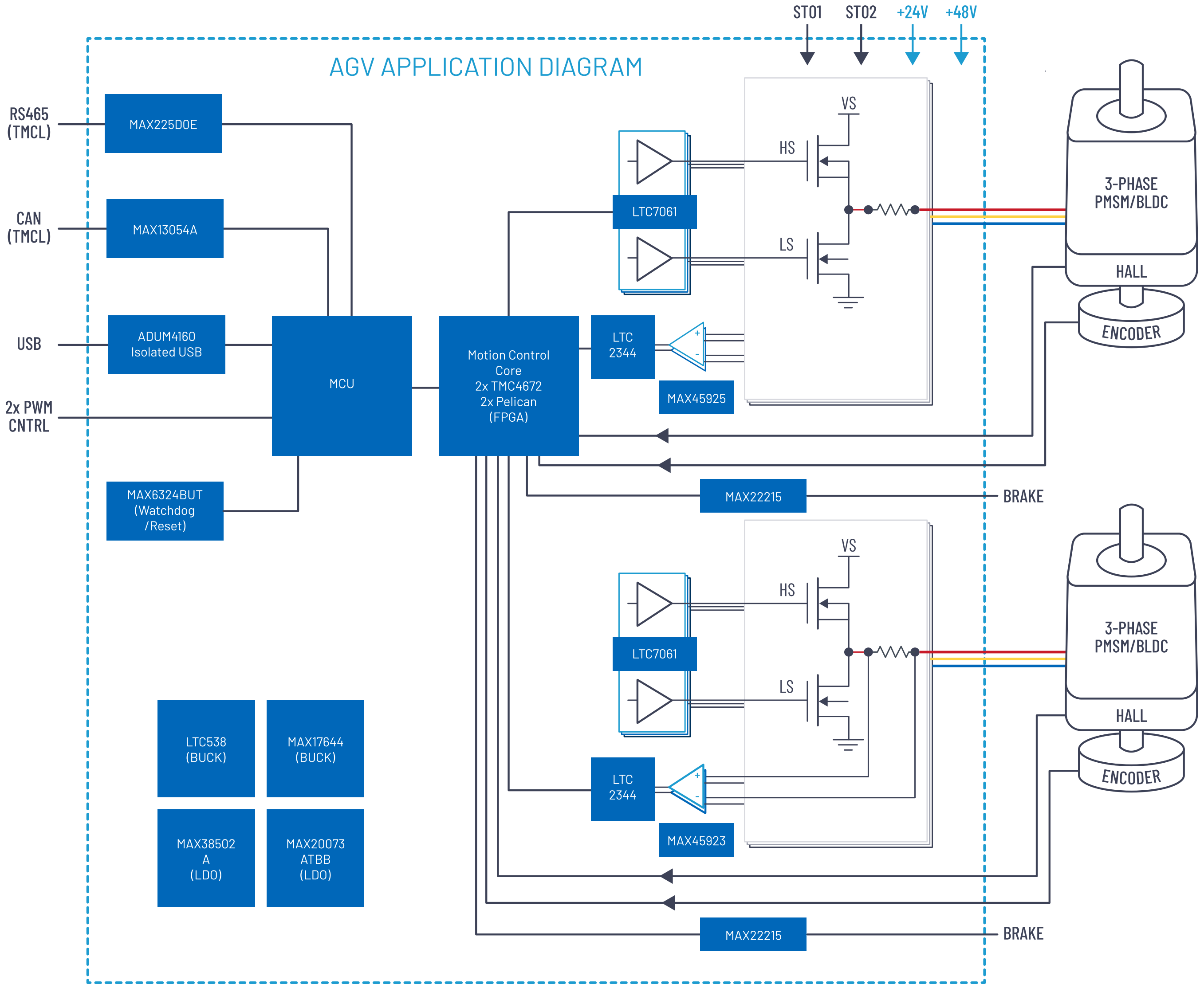


Figure 4: *TMCM-2611-AGV Block Diagram*

ENSURING SAFETY WITHIN A GROWING AUTONOMOUS MOBILE ROBOT WORKFORCE

Autonomous mobile robots (AMRs) lend themselves to modern manufacturing facilities with their ability to self-navigate through dynamic environments, transporting payloads between workstations without compromising the health and safety of nearby workers. To ensure safe operation, developers, regulators, and users must collaborate to address both technical and ethical challenges.

> The deployment of AMRs in modern manufacturing represents an evolutionary leap for flexible manufacturing systems.

Previous chapters have spoken to the technology advancements within mobile robotics. In this chapter we will explore relevant regulations and standards necessary for the design, development, and safe deployment of these innovative mobile robots in contemporary manufacturing settings.

EU MACHINERY REGULATION

The new EU Machinery Regulation 2023/1230 [i], published in the Official Journal of the EU on 29 June 2023, provides requirements for emerging technologies such as Artificial Intelligence (AI), Internet of Things (IOT) and Robotics. It has defined the term 'autonomous mobile machinery' as mobile machinery that operates in an autonomous mode, where all essential safety functions are maintained during normal operation.

The new regulation will repeal the **Machinery Directive 2006/42/EC** and becomes legally binding in all EU states on 20 January 2027. To comply, manufacturers must demonstrate product conformity with harmonized standards.

There are three types of standards: Type A – Basic Safety Standard (Principles of Design), Type B – Generic Safety Standard (Particular Safety Aspects) and Type C – Machinery Safety Standard (Safety Guidelines).

COMPLIANCE WITH HARMONIZED STANDARDS

The initial step in the compliance process is to determine the risk associated with operating the AMR in an enclosed space with obstacles in the vicinity. The manufacturer must perform a risk assessment to identify potential **hazards** and **hazardous situations** and suggest potential **risk reduction** measures.

RISK ASSESSMENT

The Type A standard EN ISO 12100:2010 [ii], "Safety of Machinery – General principles for design – risk assessment and risk reduction," provides guidance in the principles and methodology for achieving safety in the design of machinery.

For guidance on hazards associated with AMR, Annex B of EN ISO 3691-4:2023 [iii] provides a list of significant mechanical, electrical, thermal, and ergonomic hazards. This regulation outlines specific safety requirements and verification methods for driverless trucks, including Automated Guided Vehicles (AGVs), AMRs and Automated Guided Carts (AGCs).

A "regulation" is a binding legislative act. It must be applied in its entirety across the EU.	▶ Law & Directives: Machinery Directive: 2006/42/EC Low Voltage Directive: 2014/35/EU EMC Directive: 2014/30/EU	
	▶ Type A Standard: ISO 12100:2010 - General Principles for Design	**Basic** Safety Standard - Principles of Design
	▶ Type B Standard: IEC 62061:2021 - Safety of Machinery - Functional Safety ISO 13849-1:2023 - Safety of Machinery - Control Systems	**Generic** Safety Standard - Particular Safety Aspect
	▶ Type C Standard: ISO 10218-1/2:2011 - Robots & Robotic Devices ISO 3691-4: 2023 Industrial Trucks	**Machinery** Safety Standard - Safety Guideline

Figure 1: *Overview of the 3 types of Standards for EU Machinery Regulations*

RISK REDUCTION

Sub-Clause 4 of EN ISO 3691-4:2023 [iii] details the safety requirements and protective measures necessary for these autonomous mobile machines, including details about modes of operation, detection of person in the path, speed control, braking system, electrical systems, and automatic battery charging.

Figure 2 illustrates the relationship between protective measures and the hazard situations in which they apply. There are five performance levels, with PL_a offering a minimal contribution to risk reduction, and PL_e providing the highest contribution to risk reduction.

COMPLIANCE IN ACTION

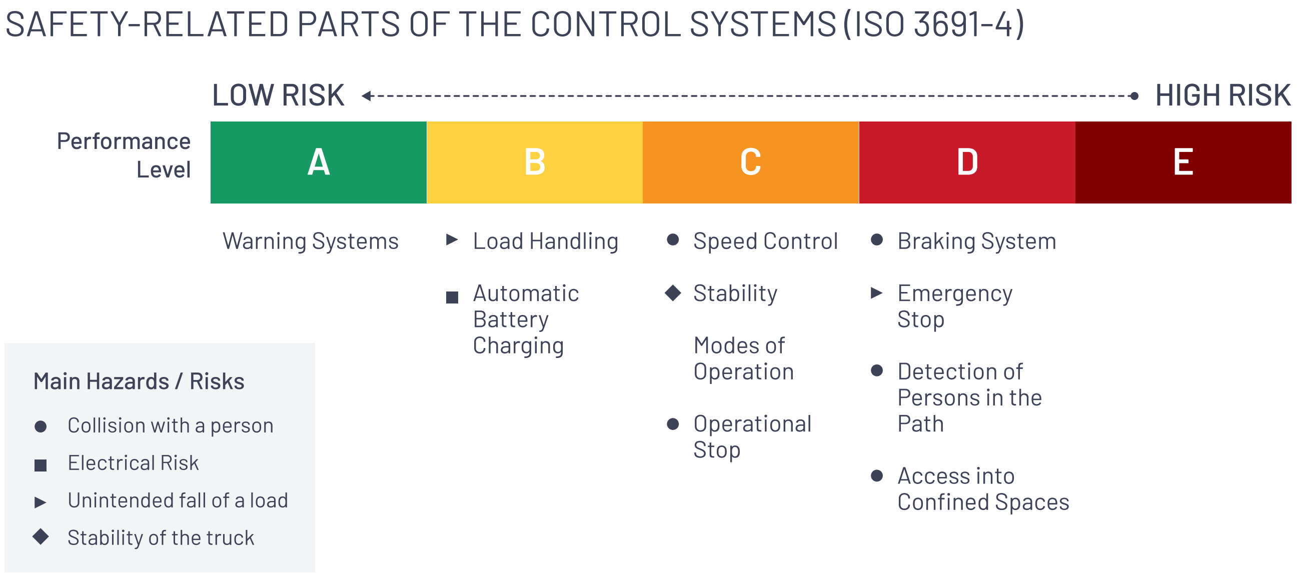
Here's an example of what it might look like to comply with the harmonized standards under EU Machinery Regulation 2023/1230.

The hazard of colliding with a person can be mitigated by detecting a person in the path of the AMR. Typical protective measures for detecting a person in the path include Electro-Sensitive Protective Equipment (ESPE) devices such as a 2D laser scanner or 3D Time of Flight (ToF) sensors.

AMRs are equipped with braking systems that, upon interruption of the power supply, ensure fail-safe operation. The braking system shall automatically activate upon detection of excessive speed and detection of person in the path.

The implementation of these protective measures, particularly the safety-related components of the control system, must meet the specified Performance Level (PL) in accordance with ISO 13849-1:2023 [iv].
The safety-related parts of the braking control system on the AMR requires minimum performance level D (PL_d) according to ISO 13849-1.

Figure 2: *Visual Representation of ISO 3691-4:2023 Subclause 4.11 – Table 3 (Safety-Related parts of the control system)*



SAFETY-RELATED PARTS OF THE CONTROL SYSTEMS (ISO 3691-4)

LOW RISK ← → HIGH RISK

Performance Level	A	B	C	D	E
	Warning Systems	▶ Load Handling ■ Automatic Battery Charging	● Speed Control ◆ Stability Modes of Operation ● Operational Stop	● Braking System ▶ Emergency Stop ● Detection of Persons in the Path ● Access into Confined Spaces	

Main Hazards / Risks

- ● Collision with a person
- ■ Electrical Risk
- ▶ Unintended fall of a load
- ◆ Stability of the truck

WHAT'S NEXT FOR SAFETY?

The deployment of AMRs in modern manufacturing represents an evolutionary leap for flexible manufacturing systems. Their autonomous capabilities and adaptability to dynamic environments can enhance productivity and provide a safe working environment for human workers.

However, the integration of these technologies requires strict adherence to regulatory standards to ensure their safe and effective operation. The recent introduction of the EU Machinery Regulation 2023/1230 and technology advancement underscores the importance of mindful design, codifying risk assessment and reduction as part of the product development process.

To learn more on industrial functional safety solutions,
visit *analog.com/industrial-functional-safety*

REFERENCES

[i] https://eur-lex.europa.eu/eli/reg/2023/1230/oj

[ii] https://www.iso.org/standard/51528.html

[iii] https://www.iso.org/standard/83545.html

[iv] https://www.iso.org/standard/73481.html

HOW THE ROBOT OPERATING SYSTEM (ROS) IS CHANGING THE GAME FOR MOBILE ROBOT DEVELOPMENTS

If you've ever wanted to develop your own robotic systems or applications, the Robot Operating System (ROS) is how you get started. ROS is a robotics middleware containing a set of diverse software libraries and powerful developer tools from drivers to state-of-the-art algorithms.

> ROS is a robotics middleware containing a set of diverse software libraries and powerful developer tools from drivers to state-of-the-art algorithms.

We briefly showed how ROS can extend a mobile robots' flexibility, specifically by allowing them to be programmed for new tasks as the need arises in a previous chapter. Let's explore in more detail how ROS makes mobile robot development a few levels easier.

HOW DOES THE ROBOT OPERATING SYSTEM (ROS) WORK?

ROS **packages** are made up of nodes, which are executable programs that perform a specific task. These nodes may communicate with each other via publishing or subscribing **messages** made available as **topics**, or as a request-response task via the use of **services**.

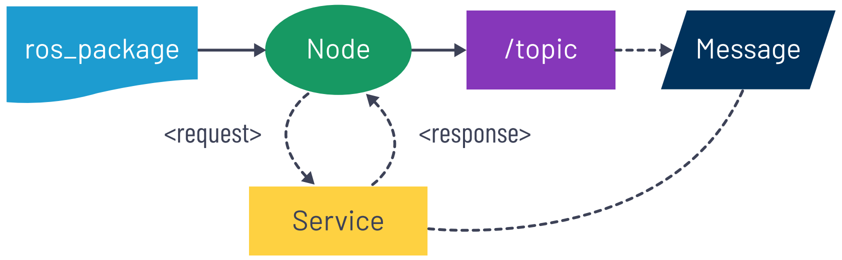


***Figure 1:** Basic ROS data flow*

TYPICAL ROS NODES IN A MOBILE ROBOT

The ROS nodes in a mobile robot are mostly drivers, wrappers, applications, or algorithms. While a **driver** can directly access the hardware, **wrappers** call on other APIs of a third-party driver that have the necessary hardware implementation access.

For example, in a mobile robot that has multiple sensors and actuators, each can have drivers/wrappers that enable robots to use their features. The processes of sensing and translating data into actions can be linked together to form different algorithms and applications.

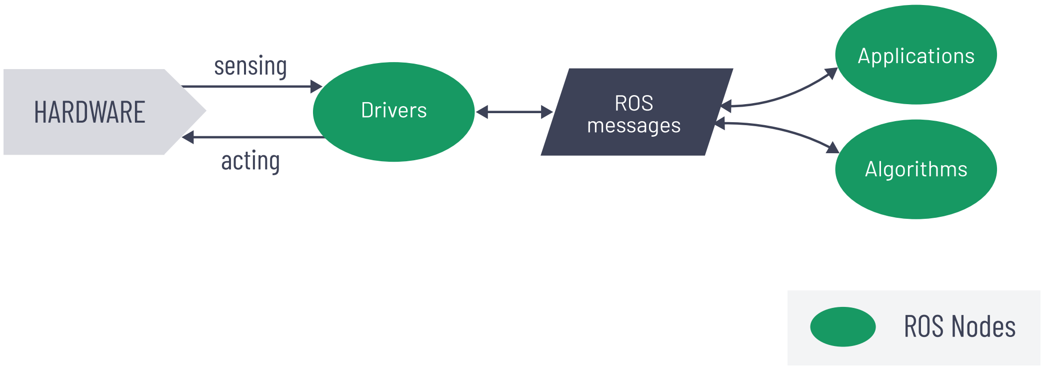


***Figure 2:** Basic ROS nodes in a mobile robot*

ROS APPLICATION FOR A SIMPLE MOBILE ROBOT

To visualize this, imagine a very simple robot that can navigate its way from Point A to Point B while avoiding obstacles. This simple robot could have (1) an IMU for localization; (2) a LiDAR to map its surroundings and avoid obstacles; and (3) a motor control to move its wheels and go to different locations. In that scenario, the ROS diagram would look like this:

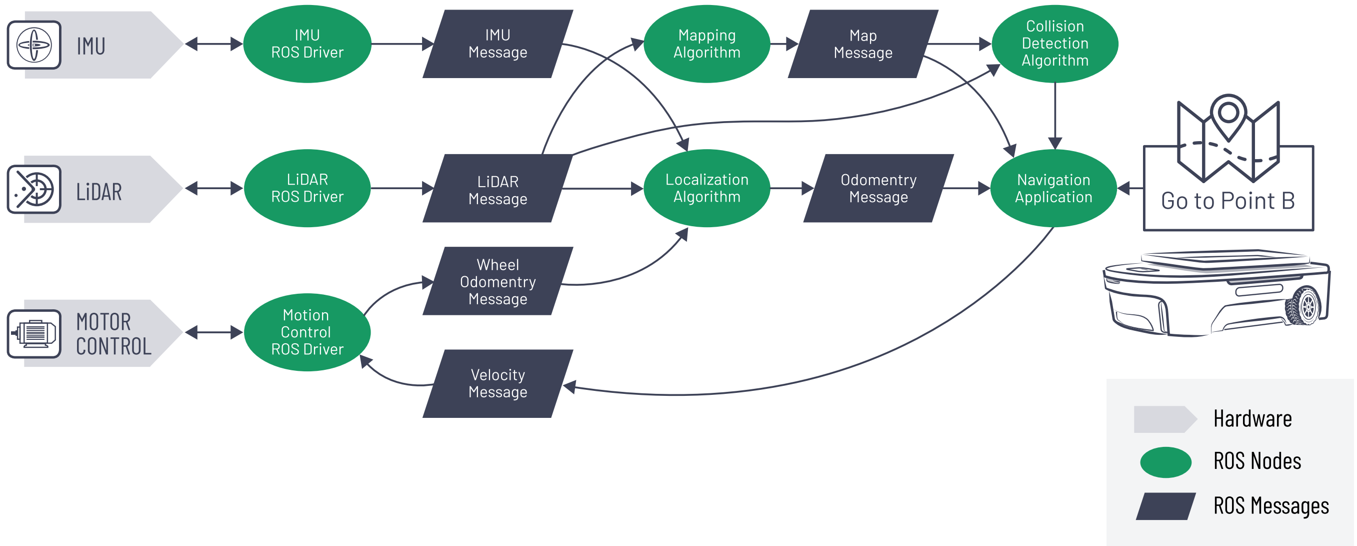


***Figure 3:** Flowchart for a ROS-based simple robot that can go from Point A to Point B while avoiding obstacles*

SO WHAT EXACTLY IS HAPPENING HERE?

- ROS drivers access hardware elements and abstract them from the algorithms and applications. In the case of the IMU and LiDAR, data is read and then broadcast as ROS messages by its corresponding ROS driver. For the motor control, its ROS driver accepts velocity messages as a "command" to move the motors and go to the intended location – however, it also has access to the wheel encoders and can output those wheel odometry messages as well.
- The localization algorithm estimates the position and orientation of a robot. First, it calculates the current position of the robot using IMU, LiDAR, and wheel odometry messages as input. Then, it broadcasts this position as an odometry message, which is picked up by the collision detection algorithm. Finally, results are cross-checked with the message from the mapping algorithm and the LiDAR message.

- **The navigation application sends a motion command via the graphical user interface**, which offers a means to control the wheels by sending a velocity message to the motor control driver. Known collision points inform the command sending the robot from Point A to Point B.

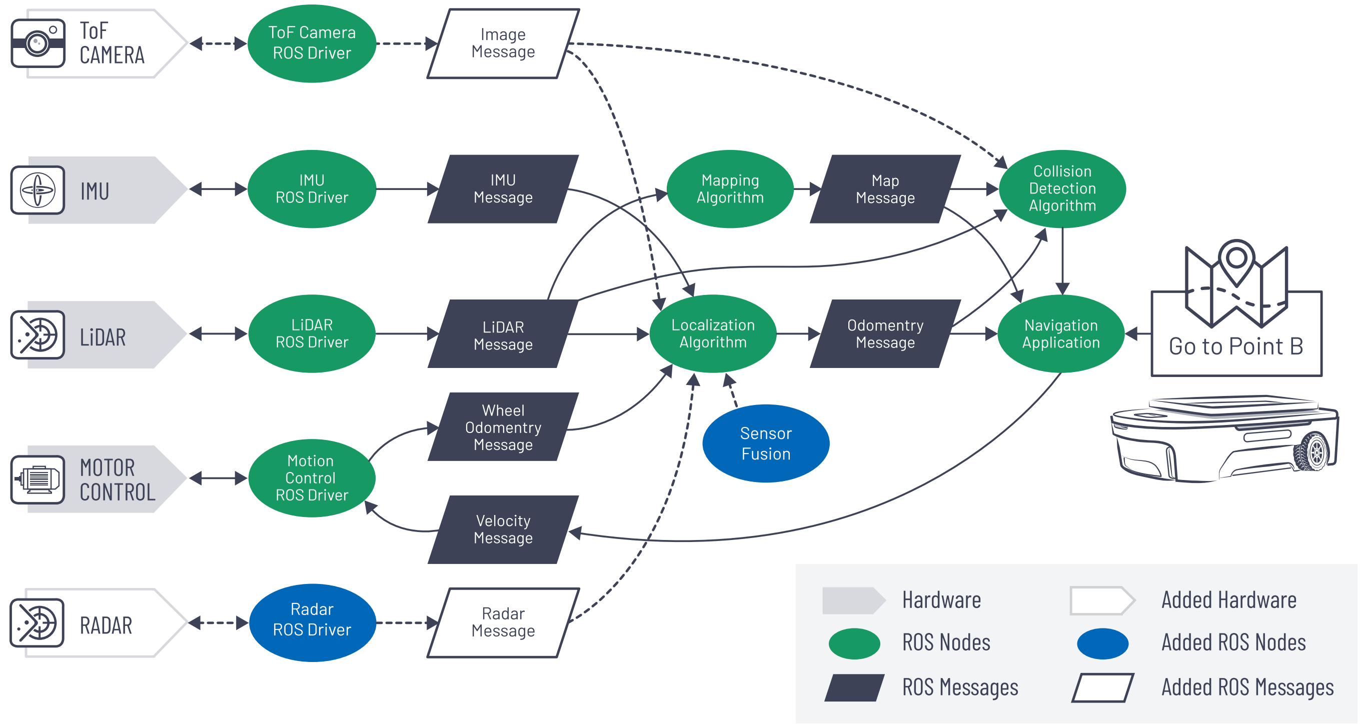
HOW ROS IS ENABLING THE CONTINUOUS EVOLUTION OF ROBOTS

ROS provides a standard message/framework that enables inter-node communication. Thus, the whole software architecture can be reused and improved by adding more ROS nodes into it, adding greater functionality to simple robots like the one above.

For example, the localization can be made more accurate by adding a sensor fusion and/or a sensor noise filtering algorithm. Or more sensors can be added, such as radar to accommodate poor lighting or time-of-flight (ToF) depth sensing cameras to enable more accurate 3D mapping and collision detection.

As long as the hardware systems have ROS drivers that make them available for use in any ROS-capable robot, the algorithms and applications that can be developed on such systems are always extensible.

Figure 4: *An evolved version of the simple robot flowchart*



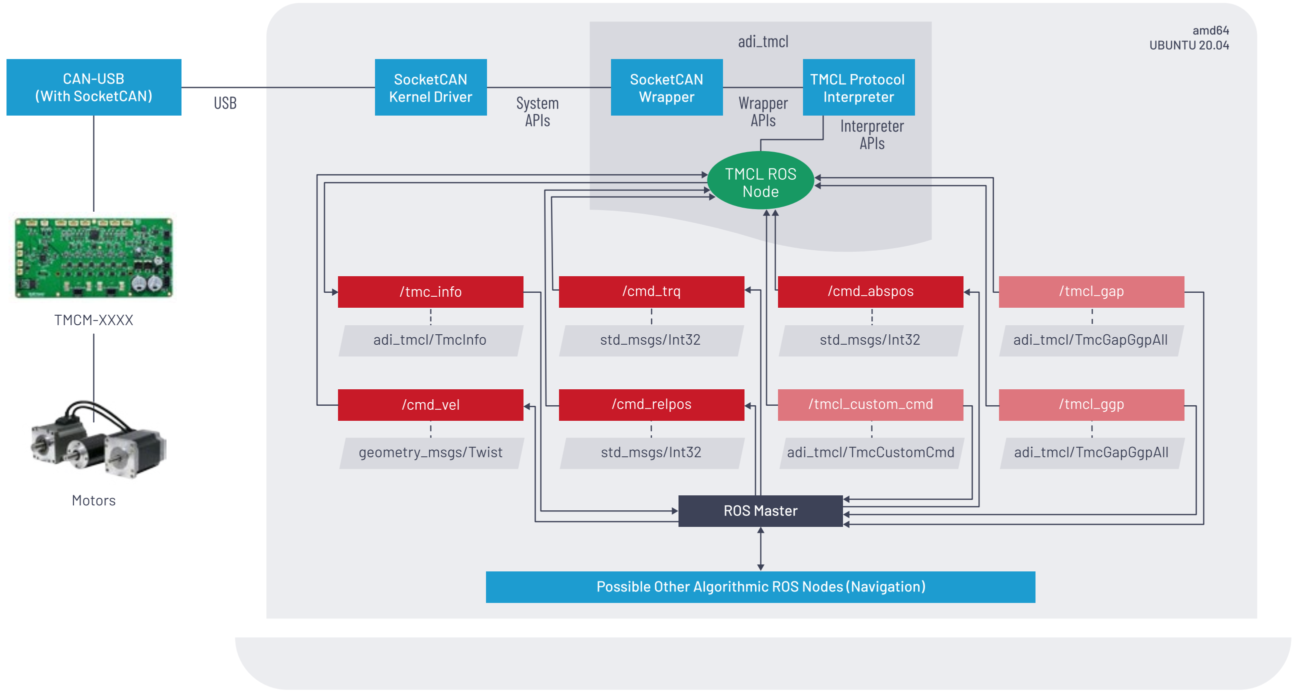
ADI TRINAMIC: MAKING MOTION CONTROL ROS-READY

We've seen how different ROS drivers can bring out sensor and actuator messages that serve as inputs and outputs of algorithms and applications within a mobile robot. Advanced technology combined with supporting ROS drivers can help developers move away from the myriad software development kits (SDKs) that are needed to make each hardware work.

This allows mobile robotics innovators to focus on evaluating new technology immediately, develop advanced algorithms and applications, while avoiding interoperability issues with third-party products.

To deliver this advantage, the ADI Trinamic portfolio combines ADI's motion control expertise with our analog process technology and power design proficiency. *Figure 5* shows a high-level software architecture of the ADI Trinamic motion control ROS driver.

Figure 5: *Software architecture of the Trinamic motion control ROS driver*



SUMMARY

ROS has applications in multiple fields (consumer, industrial, automotive, etc.) and can be deployed in a variety of platforms (Linux, Windows, MacOS, and some embedded platforms). It is completely open source with commercial options. Support for ROS is abundant due to dedicated resources across the global community, giving users an easier path for their designs and applications.

This chapter provided an overview of how ROS can be utilized in mobile robots. The next chapter will further explore building the foundations of the overall software architecture in autonomous mobile robots.

BUILDING THE FOUNDATIONS OF SOFTWARE ARCHITECTURE IN AUTONOMOUS MOBILE ROBOTS

As autonomous mobile robots (AMRs) continue to reshape industries with their capabilities in automation, navigation, and complex task execution, building a robust software architecture is paramount. This chapter will explore the essential components of the AMR software stack and highlight key considerations for developers.

> Building an efficient software architecture for AMRs requires seamless integration of hardware, middleware, software development kits (SDKs), and end-user applications.

BUILDING BLOCKS OF AMR SOFTWARE ARCHITECTURE

A software stack consists of various subsystems that work together to create a complete platform for running applications. *Figure 1* shows a simplified software architecture diagram.

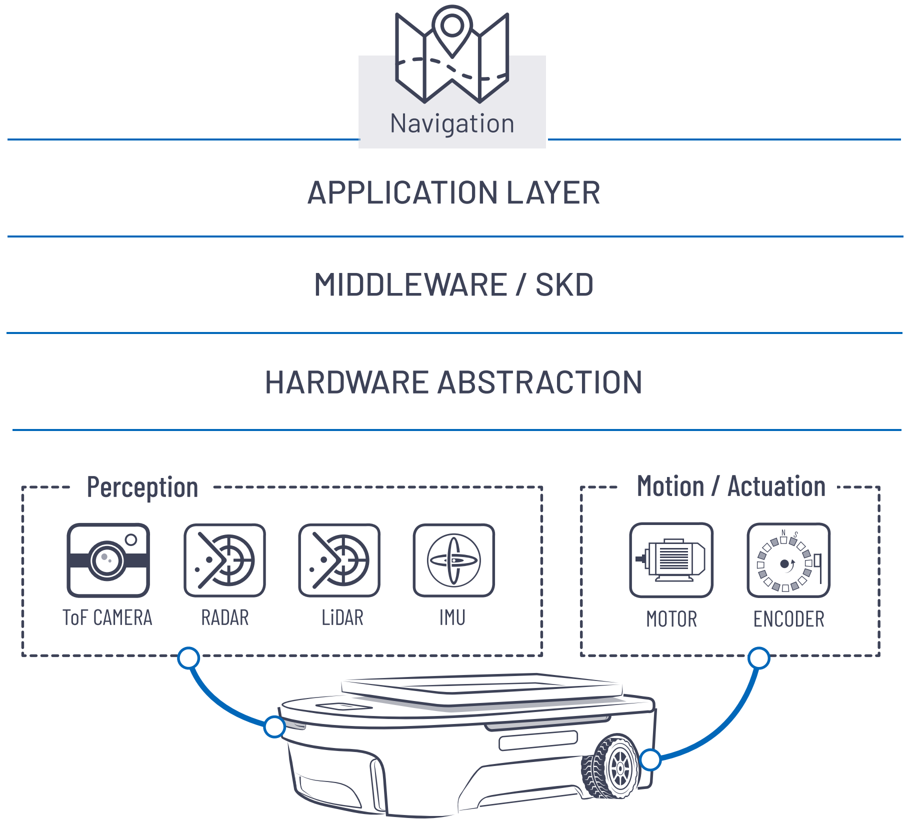


Figure 1: *Simplified AMR software architecture diagram*

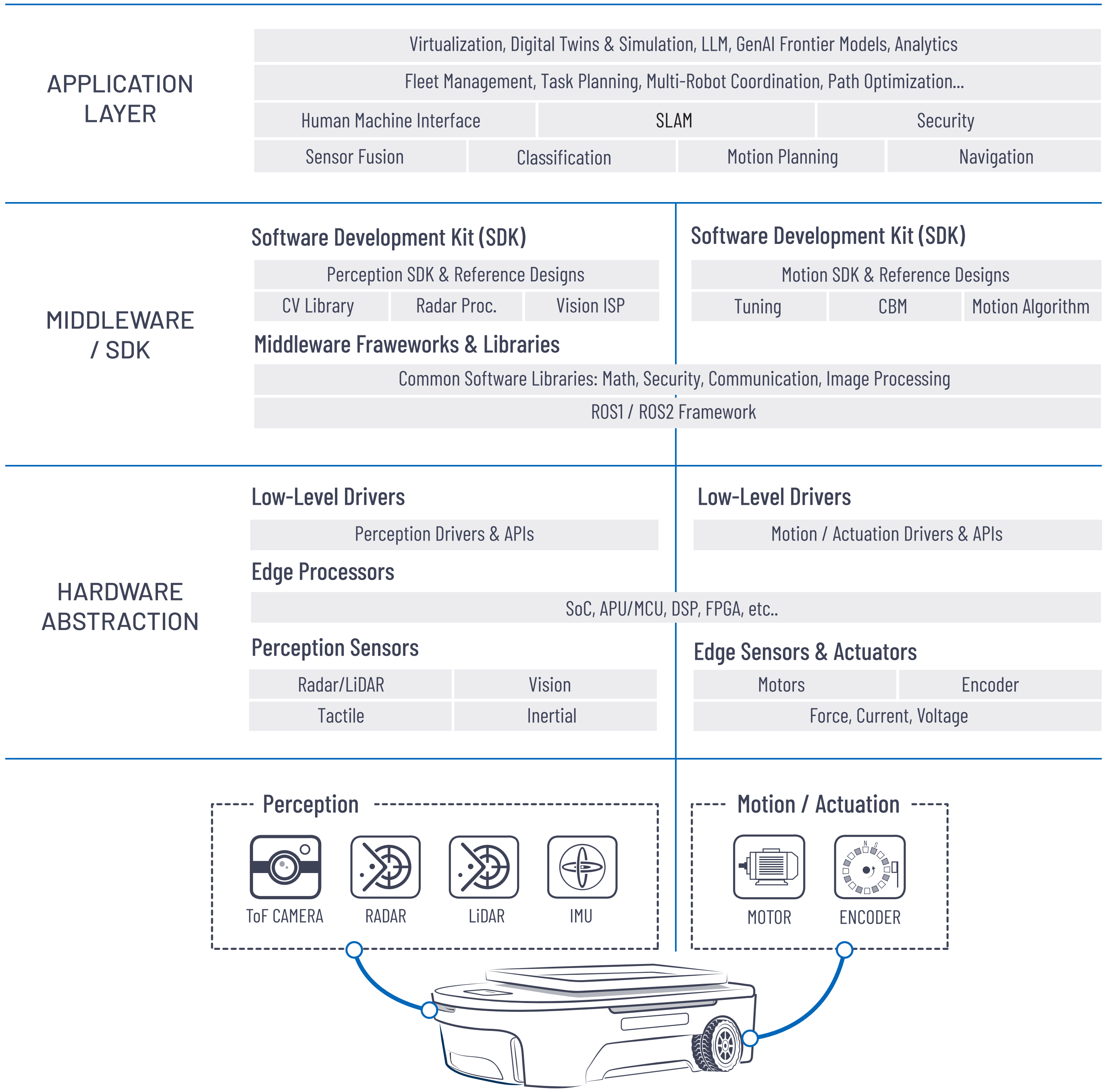
The **hardware abstraction** layer allows the operating system to interact with hardware devices at an abstract level without needing to manage the complexities of the hardware itself. As the name suggests, the **middleware** exists in the middle, acting as a bridge between the operating system and applications. Finally, the **application layer** facilitates communication with other applications and systems in the network. With these basics, it's time to dive deeper into the architecture.

KEY LAYERS OF AN AMR SOFTWARE STACK

Building an efficient software architecture for AMRs requires seamless integration of hardware, middleware, software development kits (SDKs), and end-user applications. Each layer plays a critical role in optimizing performance and enabling AMRs to operate effectively.

Figure 2 shows a more detailed breakdown of each layer and outlines how the integration of hardware and software enhances the overall performance of an AMR.

Figure 2: *Detailed AMR software architecture diagram*



HARDWARE ABSTRACTION LAYER (HAL) OR THE EMBEDDED EDGE LAYER

This layer includes low level drivers for sensor hardware such as RADAR, LiDAR, Time-of-Flight (ToF) cameras, and inertial measurement units (IMUs) that provide data for environmental awareness and motion sensing. From an actuation perspective, this layer has low-level drivers for motors and encoders that you will find, for example, within an AMR.

Sensing and actuation activities run on microcontrollers (MCU), compute systems, or systems on chips (SoC), which typically use embedded Linux or RTOS to execute real-time data processing tasks. Low-level drivers within the HAL bridge the gap between hardware and higher-level software by providing direct control over the system's sensors and actuators.

MIDDLEWARE LAYER

The robot operating system (ROS) serves as the primary middleware for open-source/prototyping and quick testing of sensor modalities in AMR development. It provides a rich set of libraries and frameworks that simplify the integration of perception, motion control, and application layers. Both ROS 1 and ROS 2 leverage Ubuntu Linux as the underlying operating system, with inbuilt support for various communication protocols such as USB, Ethernet, etc. To refresh your knowledge of ROS basics refer to the previous chapter.

In addition to ROS, the Software Development Kit (SDK) layer provides the building blocks for developing and deploying perception and motion algorithms.

APPLICATION LAYER

The application layer is where high-level functions such as object identification, route planning, and pick-and-place movements come to life. These applications depend on data processed from the lower layers of the stack, and they directly impact how the robot interacts with its environment.

For tasks like route planning and navigation, the seamless integration of perception and motion data ensures that robots can adapt to changing environments quickly. Cloud integration further enhances the robot's capabilities, extending processing beyond the device. This allows advanced functions like deep learning model training, digital twin simulations, remote monitoring, and fleet management.

BEST PRACTICES FOR SYSTEM INTEGRATORS

By offering cutting-edge sensor technologies, edge processing capabilities, and ROS 2 integration packages, Analog Devices empowers system integrators to develop AMRs that are not only intelligent and adaptive, but also reliable and scalable. Our holistic solutions and the best practices below can help developers build AMRs that are well-prepared for the demands of modern automation.

- **Leverage ROS 2 for Flexible, Scalable Development:** ROS 2 enables scalability through modularity, making it easy to adapt the system for new sensors or functionalities. Pre-built ROS 2[1] nodes are readily available from ADI for various functionalities.[2,3,4,11]
- **Optimize Real-Time Performance with Edge Processing:** Efficient edge processing can reduce latency and enhance a robot's responsiveness. ADI offers modular, real-time tuned, plug-and play-solutions for motor control[5], ToF[6], IMU[7], and BMS[8].
- **Utilize SDKs to Customize Applications:** SDKs provide tools for tailoring perception and motion algorithms, allowing developers to customize AMR performance for their target applications. ADI provides Linux-based SDKs for motor control[9], IMU[10], and ToF[11].
- **Ensure Secure and Reliable Cloud Integration:** When interfacing with cloud services, always incorporate secure communication protocols and data management practices to maintain data integrity.[12]

REFERENCES/RESOURCES

[1] Robot Operating System (ROS) Development Platforms

[2] Official ROS 2 Driver for Trinamic Motor Controllers (GitHub: tmcl_ros2)

[3] ROS 2 Bindings for Time-of-Flight (GitHub: tof_ros2)

[4] C++ ROS 2 Node that Reads Sensor Data from ADI IMU and Publishes Message to Topic (GitHub: imu_ros2)

[5] TMCM-2611-AGV Dual Axis Servo Drive Platform for 3-phase BLDC Motors

[6] EVAL-ADTF3175 Time-of-Flight Evaluation Kit

[7] ADIS16500/PCB Breakout Board

[8] ADBMS6948 16-channel, Battery Pack Monitoring System

[9] TMC Tech Access Package – Open Source IC Software API

[10] ADIS16475 IIO Inertial Measurement Unit Linux Driver

[11] ADI Time-of-Flight SDK (GitHub: ToF)

[12] Ensuring a Secure Future for Robotics: The Role of Cybersecurity

WHAT ROLE DO ALGORITHMS PLAY IN MOBILE ROBOTICS?

> Algorithms put the *autonomy* in autonomous mobile robots (AMRs).

Algorithms put the *autonomy* in autonomous mobile robots (AMRs). They help mobile robots process sensor data, detect obstacles, and make real-time driving decisions to better navigate their environment.

While algorithms are continuously evolving due to advancements in artificial intelligence (AI) and machine learning, you can broadly classify them under three categories:

- Sensing
- Actuating
- Communicating

These three capabilities give mobile robots everything they need to sense their surroundings, actuate or control their movements, and communicate efficiently between different systems within the robot. This chapter will focus on the **sensing** and **actuating** algorithm types.

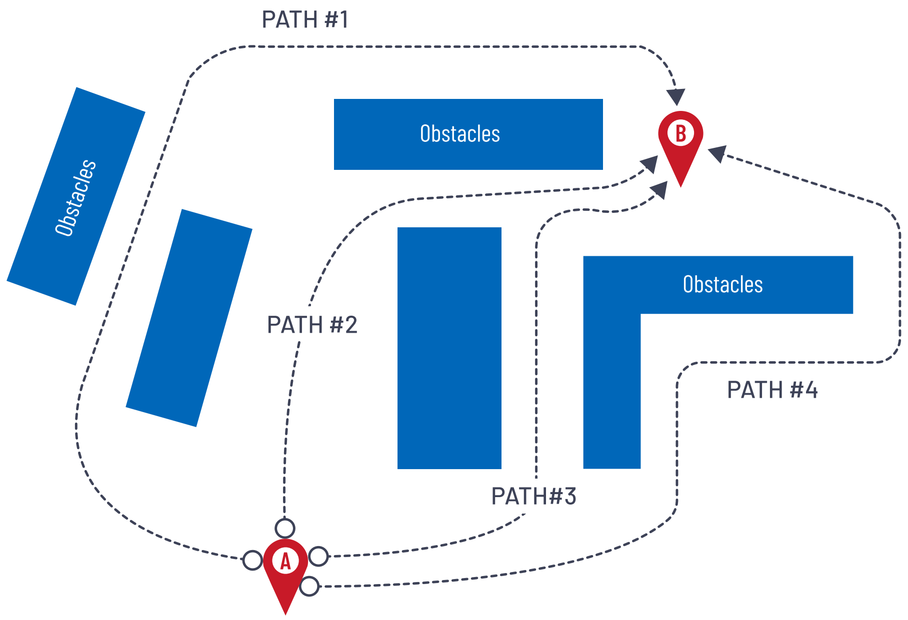


Figure 1: *Concept diagram of potential path options for autonomous vehicles*

Let's consider the scale of the task: To navigate from Point A to Point B with a potentially unlimited number of obstacles in your path. Humans use our eyes to assess the world around us, make judgments based on those assessments, and set off in our desired direction. As we proceed, we may evaluate whether a faster or shorter path is available. Mobile robots operate similarly, using sensing and actuating algorithms to make informed decisions from collected sensor data.

SENSING ALGORITHMS

Sensing algorithms help mobile robots to perceive their environment. AMRs gather data from many sources, such as time-of-flight (ToF) cameras, inertial measurement units (IMUs), and lasers. Sensors provide autonomous mobile robots with an accurate and detailed understanding of their surroundings using algorithms such as:

- Simultaneous Localization and Mapping (SLAM): LiDAR-based SLAM algorithms help robots build maps of their surroundings and determine their location within the map. Techniques like **Iterative Closest Point (ICP)** and **graph-based SLAM** can be used.
- Point Cloud Processing Algorithms: ToF and LiDAR technology generate Point Clouds, which represent the detected surfaces of objects. The resulting 3D models can be used for segmentation, clustering, and object detection.
- Object Detection Algorithms: For spatial awareness tasks like obstacle detection and avoidance, ToF sensors utilize a range of algorithms, from connected-component labeling to advanced deep learning algorithms.
- Sensor Fusion Algorithms (ex. Complementary Filter, Kalman Filter, and Extended Kalman Filter): Sensor fusion algorithms combine data from accelerometers, gyroscopes, and magnetometers to provide accurate estimates of orientation, position, and velocity. The Kalman Filter, for instance, is commonly used for real-time sensor fusion because of the advantages it offers in navigation and stabilization.
- Attitude and Heading Reference Systems (AHRS): AHRS algorithms build on IMU data to calculate precise orientation and heading, which is crucial for drones, self-balancing robots, and navigation.
- Human-Robot Interaction: In challenging lighting environments, you may implement a vision sensor such as ToF to enable gesture recognition. If the environment is quiet, key phrase detection may be used. Both communication interfaces can be implemented by running a deep learning inference.

ACTUATING ALGORITHMS

Actuation focuses on controlling the movement of robotic components such as wheels, arms, or grippers. Naturally, precision is a priority, as these algorithms enable robots to carry out complex movements and interact with their environment in real time. Some examples are:

- Proportional-Integral-Derivative (PID) Control ensures smooth, stable, and accurate motion control by minimizing errors between desired and actual states.
- Model Predictive Control (MPC) predicts the robot's future state based on its current state and uses optimization techniques to determine the best set of actions over a given time. This is advantageous in dynamic and complex environments.
- Trajectory Generation Algorithms compute smooth, collision-free paths for robotic motion. Examples include polynomial trajectory planning and minimum jerk paths.

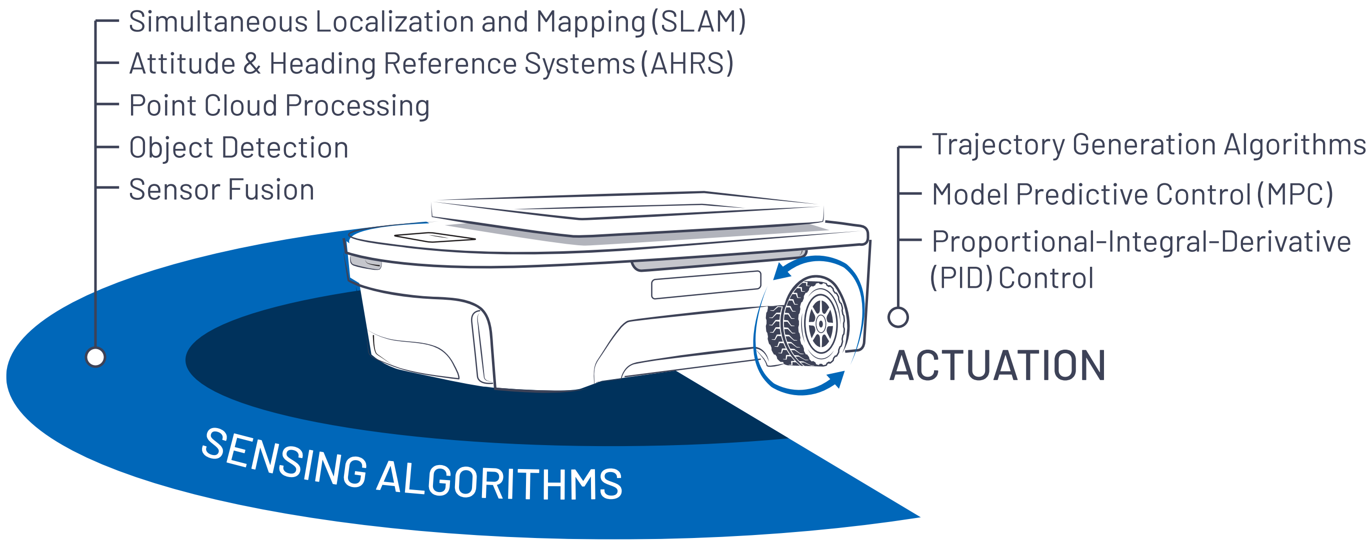


Figure 2: *Sensing and actuation algorithm options for autonomous vehicles*

MOBILE ROBOTICS ALGORITHMS IN CONTEXT

The algorithms discussed here leverage the output of hardware devices and deliver more complete solutions in robotics applications. Each algorithm serves a specific purpose within the overall robotic design and has allowed autonomous mobile robots to adapt and evolve in a dynamic and complex environment.

IMUs, Hall sensors, depth computing engines, and motion control ICs have sophisticated algorithms built into them, making it possible to run other algorithms on top of them and achieve greater speed and power performance. As algorithms advance, the speed and ease of operating robots will continue to improve.

TOP 3 TRENDS FOR INDUSTRIAL MOBILE ROBOTICS IN 2025 AND BEYOND

It took 75 years for the telephone to reach 100 million users and only two months for ChatGPT to reach the same. You won't be surprised to hear Artificial Intelligence (AI) is a growing trend within industrial mobile robotics — but it's certainly not the only trend.

We began this ebook with a chapter discussing how the automation of tasks is becoming commonplace in all facets of our lives. It seems fitting as we close out to discuss what the future holds for industrial mobile robotics and what we can expect to see in the coming years.

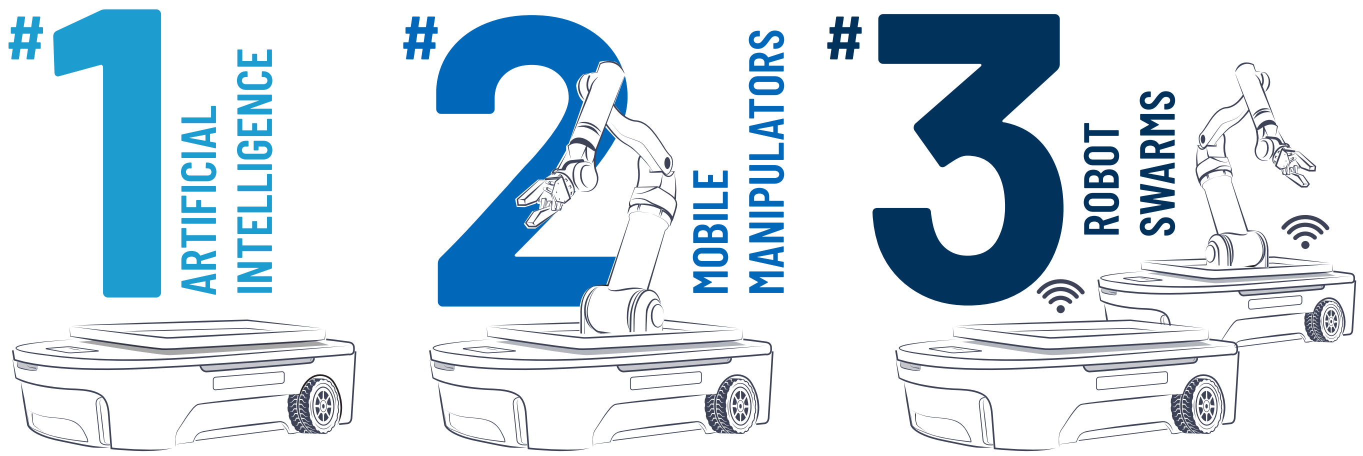


Figure 1: *Top three trends in industrial mobile robotics*

> Robots are the icon of the digital era and the most important interface between AI and the physical world.

#1. GREATER AUTONOMY THROUGH AI

Robots are the icon of the digital era and the most important interface between AI and the physical world. Machine Learning and Generative AI are creating cognitive (or real-time aware) robots that learn from their environment, make autonomous decisions, and adapt to changing scenarios.

Robots are becoming self-learning, which helps them respond to dynamic environments or implement corrective action. They can even enact or re-enact human behaviors. In the 1990s, robots had insect-grade intelligence. Today, AI has unlocked full autonomy for robots, making them safe to interface in many human environments.

With greater intelligence moving to the edge and with ultra-low-power convolutional neural network accelerators in the market already, there is growing potential for AI engines to run on localized systems. This could unlock more efficiencies that will take automation to new heights.

Finally, with the growing acceptance that AI is all around us and is used in many sectors, industrial processes and systems like mobile robotics are primed for AI-expanded capabilities.

> Future robotic innovations will require more system integration, so closer collaboration between robot developers and customers will become essential.

#2. MOBILE MANIPULATOR ROBOTS

Autonomous mobile robots (AMRs) have proven their worth, and their next evolution is already on its way. Mobile manipulator (MoMa) robots combine the mobility of an AMR platform with the dexterity of a mounted robotic arm, bridging the gap between mobility and automation.

Improved sensory perception is enabling MoMas to become part of the robotic workforce. They have excellent accuracy and precision, allowing them to perform inspections and carry out maintenance tasks with the aid of advanced sensor and camera technology. Mobile manipulators will no longer just be the vehicle that connects disparate parts of the manufacturing process; they will transport material, inspect operations, and maintain systems around the clock.

With continued advancements in algorithm developments for path planning, obstacle detection and collision avoidance, the sky is the limit for what these robots will be capable of in the coming years. Move over AMRs, here come MoMas.

#3. ROBOT SWARMS

"Many hands make light work," as the saying goes, and it's just as true for robots as it is for people — a novel and transformative concept in AMRs. A swarm of smaller mobile robots working together can accomplish more than ever before.

Seamless real-time communication is the key to the swarm's success. To collaborate on tasks that require precision handoffs or parallel work, robots must be able to send and receive information amongst themselves without any latency. Messages received out of order (or, worse, lost in the ether) can disrupt the entire operation.

As communication protocols advance, the merging of information technology (IT) and operational technology (OT) networks is enabling the potential for robot swarms. Edge-to-cloud connectivity, near real-time communication, and hybrid networks of wired and wireless technology are pushing the boundaries of what mobile robots can achieve in process automation.

KEY TAKEAWAYS

The ecosystem of how mobile robots are deployed, developed, and designed is evolving alongside the technologies above, making the future of mobile robotics smarter and more versatile than ever. Future robotic innovations will require more system integration, so closer collaboration between robot developers and customers will become essential.

MOTION CONTROL

Dual Axis Servo Drive Platform for 3-phase BLDC motors	TMCM-2611-AGV
Single Axis FOC Servo Controller Gate Driver Module	TMCM-1690
Servo Drive for 3-phase BLDC motors with RS-485 Interface	TMCM-1640

LOCALIZATION & NAVIGATION

Precision, Miniature MEMs IMU	ADIS16500
Wide Dynamic Range, Miniature MEMs IMU	ADIS16470
1 MegaPixel Time-of-Flight Module	ADTF3175
Time-of-Flight Evaluation Kit	EVAL-ADTF3175

POWER MANAGEMENT

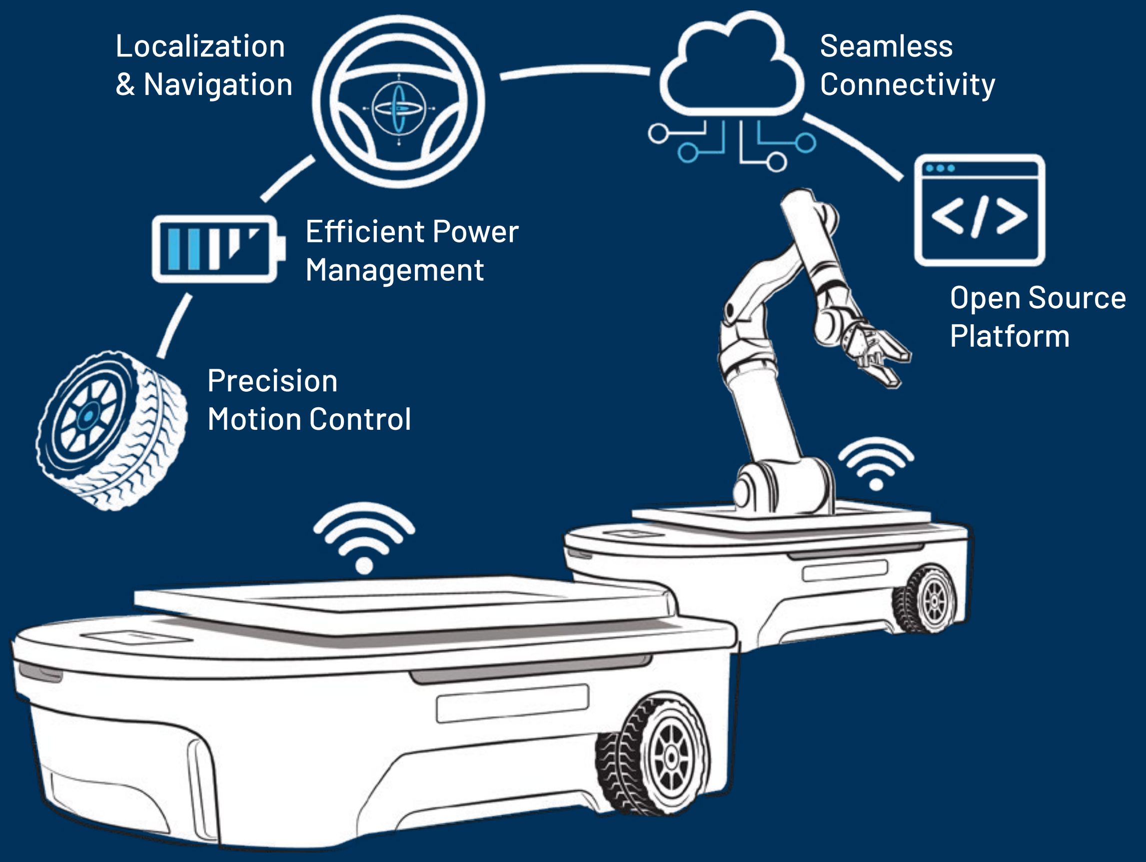
16-Channel, Battery Pack Monitoring System	ADBMS6948
18-Cell Battery Monitor with Daisy Chain Interface	ADBMS1818
High Efficiency, 140 V 250 mA Step-Down Regulator	LTC3638
High-Efficiency, Synchronous Step-Down DC-DC Converter	MAX17644
Single 2 A/3 A 2.2 MHz Low-Voltage Step-Down DC-DC Converters	MAX20073
10.5 μV_{RMS} Low Noise 500 mA LDO Linear Regulator	MAX38902A

CONNECTIVITY

MAX22500E	100 Mbps Half-Duplex RS-485/RS-422 Transceivers for Long Cables
MAX13054A	+5 V, 2 Mbps CAN Transceiver with ±65V Fault Protection
ADuM4160	Full/Low Speed USB Digital Isolator
ADMV9615 / ADMV9625	60 GHz Millimeter-wave Short Data Link
MAX96717 / MAX96724	GMSL Serializer/Deserializer
ADIN1200 / ADIN1300	10 Mbps, 100 Mbps and 1 Gbps Industrial Ethernet PHYs
ADIN3310 / ADIN6310	3/6 Port Industrial Ethernet Time Sensitive Networking Switch

SOFTWARE DRIVERS

tmcl_ros (GitHub)	ROS Driver for ADI Trinamic Motor Controllers (TMC)
tmcl_ros2 (GitHub)	ROS2 Driver for ADI Trinamic Motor Controllers (TMC)
tof-ros2 (GitHub)	ROS2 Bindings for Time-Of-Flight
ADI Time-of-Flight SDK (GitHub)	Cross Platform Library for ADI Depth Cameras
imu-ros2 (GitHub)	ROS2 C++ node for IMU device configuration
ADIS16xxx (wiki)	Linux Industrial I/O (IIO) subsystem driver for IMUs



ANALOG DEVICES

VISIT ANALOG.COM/MOBILE-ROBOTICS