SunCam online continuing education course

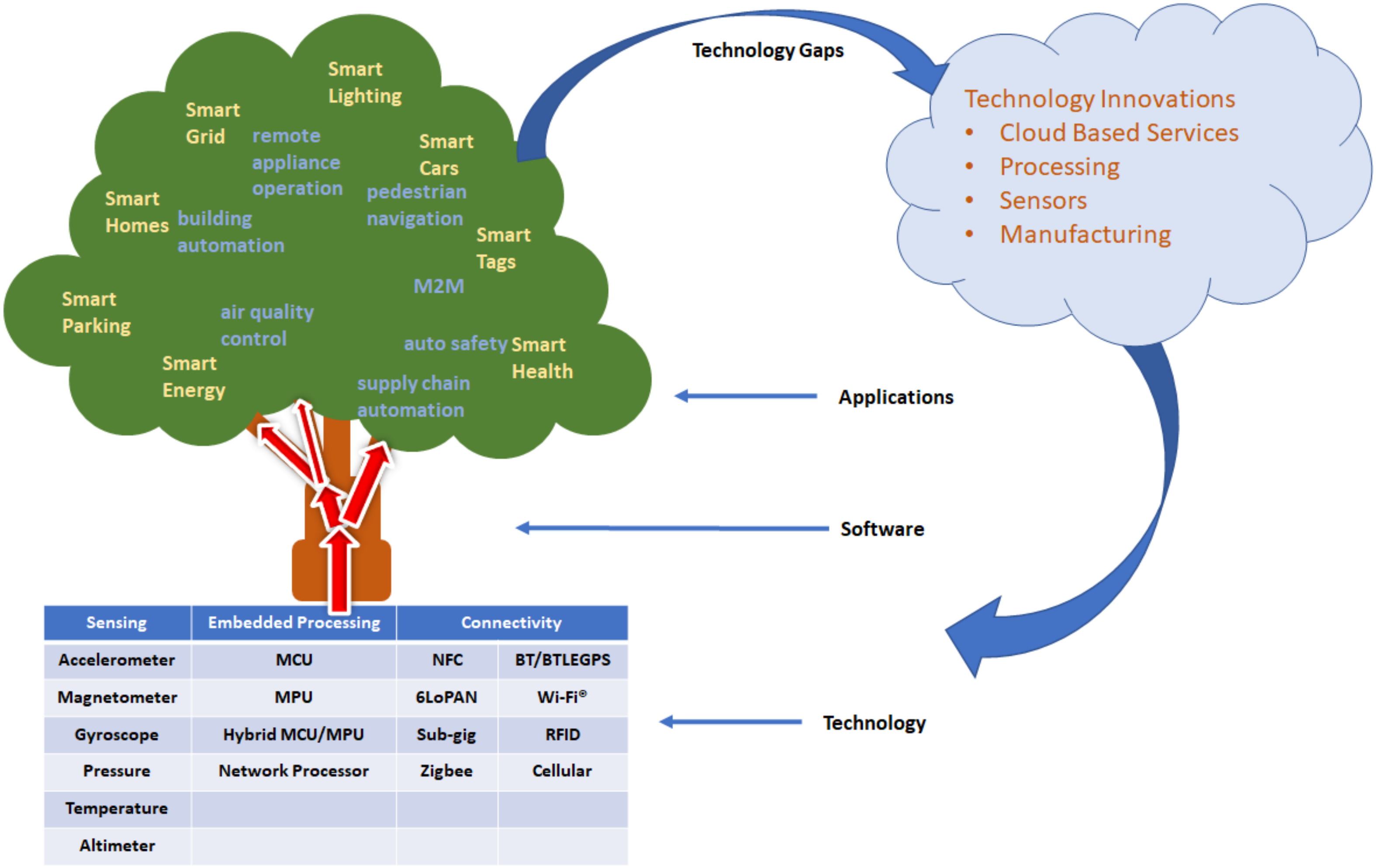
Understanding Sensors Part 2

Sensor Networks

by

Peter Kennedy

1.0 Overview: Part 1 of the course covered the sensing system, sensor performance, and descriptions of several different sensor types. Part 2 begins with a description of two additional sensors that will continue to play an important role in the evolving sensor networking technology; micro electro-mechanical systems (MEMS) and fiber optic sensors. Then sensor networking architecture and the transmission of measured data to a sensor fusion algorithm is examined. Finally, sensor fusion technology is described. Sensor fusion enhances the information available on the quantities of interest as well as the interactions between them. The human body is good example of a networked sensing system that performs sensor fusion allowing us to make decisions. Body sensors allow us to see, smell, taste, hear, and feel. These sensors are effectively networked to a central processor; the brain. The brain assimilates and fuses all the sensor data; evaluating it relative to a goal, function, or action we want to do and then applies a decision process to help us choose a path forward. This part of the course will cover all aspects of the networked sensing system; the sensor node, the communication network, network topologies and wireless sensor networks (WSN), communication network layered protocols, and finally fusion algorithms and processing techniques, many of which are now available on the Cloud. Network connectivity, person to person, machine to machine, device to device and combinations thereof is growing at a fast rate. Spurred by the developments in the smart phone, the Internet, MEMS, the Internet of Things (IoT) and Cloud, devices are becoming smart and capable of user control and monitoring from remote locations. Figure 1.0 [1] illustrates the IoT connectivity and user activity supported by sensor connectivity and fusion technology, with technology gaps fed back as requirements for the development cycle. The diversity of applications also poses some unique design issues relative to data prioritization. Needing information on a product's availability or the location of a parking space in a garage; while important does not have the same level of criticality that a bridge integrated with stress sensors indicating the possible occurrence of failure condition has. Data priority needs to have an associated level of criticality or possibly special links or even networks required for certain life critical applications. Information is the driving factor for decision making and interfacing with IoT sensor connectivity can enhance our quality of life, providing daily living improvements to all aspects of it. But responsibility for dissemination of this data accompanies this functionality, data priority must account for application diversity and the need for network security is tantamount to protect user data privacy and provide for a cybersecure information storage base.



Sensing	Embedded Processing	Connectivity	
Accelerometer	MCU	NFC	BT/BTLEGPS
Magnetometer	MPU	6LoPAN	Wi-Fi®
Gyroscope	Hybrid MCU/MPU	Sub-gig	RFID
Pressure	Network Processor	Zigbee	Cellular
Temperature			
Altimeter			

Figure 1.0 Internet of Things Connectivity [1]

2.0 MEMS and Fiber Optic Sensors: MEMS and Fiber Optic Sensors are described in this section; both play an important role in sensor networking.

2.1 MEMS [2, 3, 4]: MEMS technology provides the foundation for many solid-state sensors. Applications are growing at a fast pace, coincident with technology advancements resulting in significant performance improvements. MEMS are used to measure quantities that include pressure, temperature, strain, force, rate and displacement, acceleration, and altitude. There are many types including accelerometers/inclinometers, oscillators, inertial rate sensors, full inertial measurement units and navigation systems, microphones, pressure, proximity, and temperature sensors, electronic compass, magnetometers, and micromirrors. MEMS motion sensors are used in the automotive industry. Pressure sensors, accelerometers and gyroscopes are in many medical and consumer electronics products such as smart phones and consoles/controllers. The foundation for MEMS technology is imbedded in the development of integrated circuit technology, and advancements in semiconductor device fabrication that have provided the means to implement mechanical motion components on a microscopic scale. A MEMS sensor can now include sophisticated signal processing and communication interfaces, such as Wi-Fi, making them a perfect match for WSN. MEMS sensors incorporate small micro-scale mechanical structures that are integrated with electrical and mechanical components in a single chip to produce a fully functional electro-mechanical sensor. The key to any MEMS sensor is a small mechanical element that can change shape, rotate, or vibrate. They collect environmental data; measuring mechanical, thermal, biological, chemical, optical, or magnetic state of physical quantities. Advances in MEMS fabrication processes led to the development of microsystems and smart sensors. Commercial MEMS devices are integrated with micro sensors, micro actuators and microelectronic ICs that sense and control the physical environment as well as communicate with each other. They constitute a control system for micro-devices such as microvalves and micropumps routing gas and liquid flow, optical switches to modulate light, and mirrors to direct light beams. Mechanical MEMS sensing techniques include piezoresistive, piezoelectric, or capacitive as described in the following paragraphs.

- Piezoresistive sensors use the piezoresistive effect; a semiconductor material element that will change its resistivity when a strain is applied to it such as silicon which is often used to make this type of strain sensor.
- Piezoelectric sensors use the piezoelectric effect; a strain applied to a piezoelectric crystal, such as quartz, generates a potential voltage difference across it and this also reciprocates such that a potential difference applied across a crystal produces a displacement or strain. This effect can be used to implement a mechanical stress sensor or actuation for small displacements.

- Capacitive sensors, described in Part 1 of the course, can provide very accurate measurements and are used for precision sensing mechanisms. They use the offset geometry between fixed conducting plate(s) and moving conducting plates, relying on the basic parallel-plate capacitor equation:

$$C = \varepsilon_r \cdot \varepsilon_0 \cdot \frac{A}{d}$$

where: ε_0 = permittivity of free space = 8.854×10^{-12} Fm^{-1}, ε_r = relative permittivity of material between the plates, A = overlapping plate area (m^2), d = plate separation (m). As capacitance is inversely proportional to the distance between the plates, sensing of very small displacements is extremely accurate.

Brief descriptions of the types of mechanical sensors that use these techniques include:

Strain gauge - the piezoresistive semiconductor element is fabricated on or bonded directly to the active surface used to measure the stain. As it deforms a strain is applied to the semiconductor material changing its resistance. The MEMS polysilicon piezoresistive strain gauge sensor is small enough to be implanted in the body to measure forces in the heart and brain tissue.

Accelerometer - accelerometers sense the external acceleration applied to a suspended proof mass producing a force (F=ma) that accelerates the mass resulting in a displacement measured by piezoresistive or capacitive techniques. Another configuration uses polysilicon springs suspending a MEMS structure above a substrate. When the body of the sensor moves in the X and Y axes, there is an acceleration causing deflection of the proof mass from its center position which is measured on four sides by differential capacitor pickoff elements.

Gyroscope – a gyroscope is a device that measures inertial angular rotation rate as described under inertial sensors. It is used in transportation, platform line of sight stabilization, navigation and missile guidance applications. MEMS gyroscopes typically use vibrating structures, beam, disk, or shell because of the difficulty of micromachining rotating parts with sufficient useful mass. They rely on measuring the Coriolis force on a body in a rotating frame. This force is applied to a mass translating (i.e., vibrating) in a rotating system, acting orthogonal to the direction of translational motion and the rotation axis. Mathematically the force is proportional to the cross product of the angular velocity vector with the translating velocity vector or;

$$F_C = -2 \cdot m \cdot (\omega \times v)$$

Where m-mass, ω-angular velocity vector, v-translating velocity vector. A mass on a rotating body (gyro mounted on a rotating platform) that is translating (vibrating) within the gyro and rotating body will exert this force on the mass support structure. Knowing the mass velocity, the rotation rate can be derived from the measured force. There are several implementations of the vibrating structure gyroscope, defined by the Institute of Electronic and Electrical Engineers (IEEE) as a Coriolis Vibratory Gyroscope (CVG), that include the Cylindrical resonator gyroscope (CRG), vibrating wheel gyroscope, Piezoelectric gyroscopes, Tuning fork gyroscope, and Wine-glass or hemispherical resonator gyroscope.

Pressure sensor – Many MEMS pressure sensors have thin membranes with sealed gas or vacuum-filled cavities on one side of the membrane with the pressure to be measured applied to the other side of the membrane. The deflection of the membrane is measured by piezoresistive and capacitive elements in contact with the membrane. Pressure sensors were among the first types of MEMS fabricated with early versions using piezoresistive strain sensing elements that were diffused into a micromachined silicon diaphragm fused to a silicon backplate and exposed to the environment through a port. An applied differential pressure caused the diaphragm to deform, being sensed by the piezoresistive elements and converted to an electrical output signal.

2.2 Fiber Optic Sensors [5]: Fiber optic sensing technology is used in the design of many sensor applications. Some advantages of the technology are they are non-electrical and explosion proof sensors. They usually do not require contact and the sensor can easily be remoted from the quantity measured. They are also easily used in inaccessible areas being small in size and weight. They can be installed in most environments and are not suspectable to electro-magnetic interference or radio frequency interference. They have high accuracy and solid-state reliability. They can interface directly to data communication systems providing secure data transmission.

Many physical quantities can be sensed optically via fiber. Light intensity, position displacement, temperature, pressure, rotation, sound, strain, magnetic and electric fields, radiation, flow, liquid level vibration. Fiber sensor sensing techniques can be categorized into three types, intensity modulated, phase modulated and wavelength modulated. Intensity modulated are generally used in displacement sensing or some other physical perturbation that interacts with the fiber or a mechanical transducer attached to it. The perturbation causes a variation in the light intensity due to the quantity be measured. Phase modulated compare the phase of light in the sensing fiber with that in a reference fiber and is effectively a device called an interferometer. Phase difference can be measured with high sensitivity and these sensors have much greater accuracy and more dynamic range sensors than those using the intensity modulation approach. Wavelength modulated sensors experience a wavelength shift associated with displacement, temperature or the presence of a chemical which causes florescence.

Intensity modulated sensors detect light intensity variations as a function of the perturbing environment. The sensor configuration is illustrated in Figure 2.0. Light loss is most often associated with transmission, reflection, or micro-bending although other phenomena are possible. Intensity modulated sensors normally require more light than phase modulated sensor designs as a result they use large core multi-mode fiber or fiber bundles. In essence the sensor is a displacement sensor. Examples of using fiber optics to detect position displacement as related to the movement of a physical property include the movement of a diaphragm for pressure or a bimetallic element for temperature (see Part 1 section 4.1.1 Thermostat).

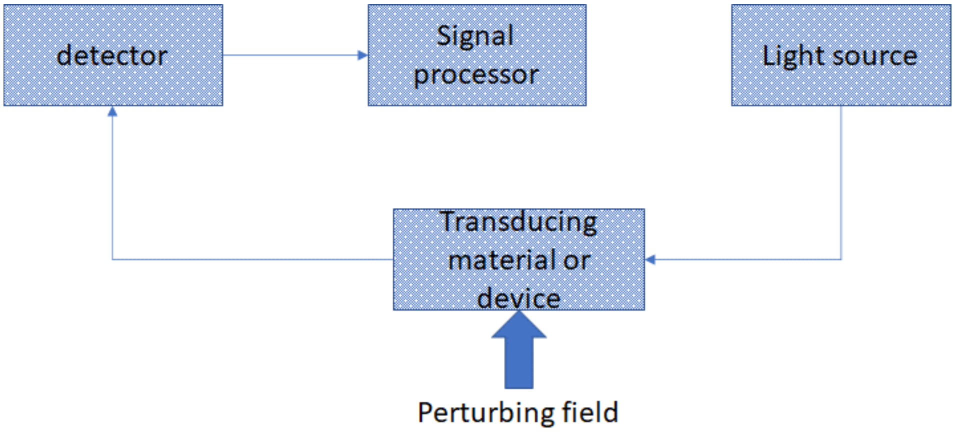


Figure 2.0 Intensity Modulated Fiber Sensor for Light Detection

Phase modulated sensors use interferometric techniques (phase interference) to detect pressure, rotation, and magnetic field; with the first two applications being the most prevalent (see Part 1 section 4.5.4 Ring Laser Gyro, and 4.5.5 Fiber Optic Gyro). Figure 3.0 illustrates the sensor configuration, termed a Mach-Zehnder interferometer, for this type of fiber sensor design. The laser light source has its outgoing beam split such that light travels down the path of a reference single mode fiber and the sensing fiber which is exposed to the perturbing environment. If the light in the reference fiber and sensing fiber are exactly in phase they constructively interfere with an increase in light intensity. If they are out of phase, they destructively interfere and light intensity decreases proportionally to the phase difference. Such devices experience a phase shift if the sensing fiber has a length or refractive index change or both when influenced by the perturbing environment.

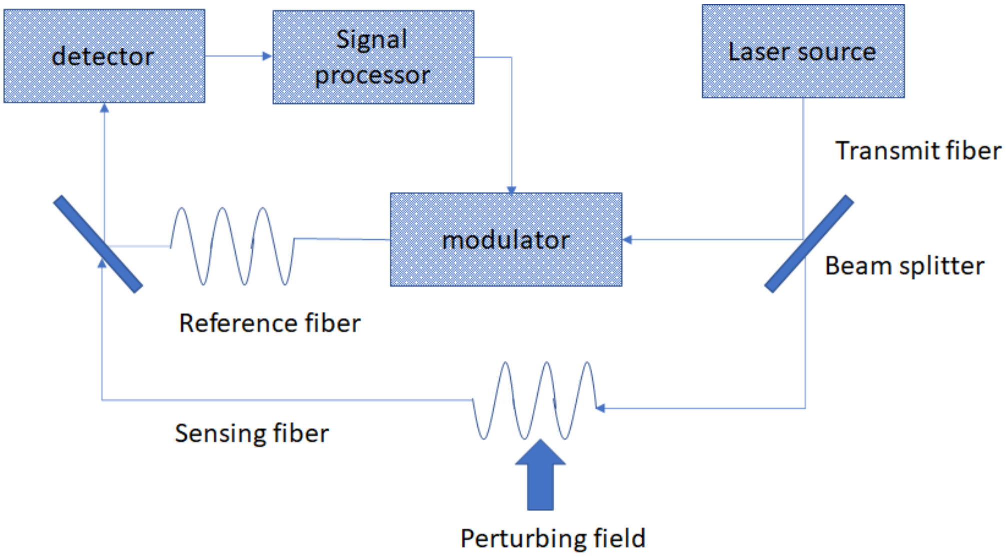


Figure 3.0 Phase Modulated Fiber Sensor for Light Detection

Wavelength modulated sensors can result from florescence but is most often implemented using wavelength modulation with Bragg gratings. Figure 4.0 shows a schematic representation of a Bragg grating based sensing system. The Bragg grating is constructed in a short segment of optical fiber which reflects specific light wavelengths and transmits all others. This is achieved by creating a periodic variation in the refractive index of the fiber core which then acts like a wavelength-specific dielectric mirror reflecting specific wavelengths of light; termed the resonance condition. A change in strain or temperature perturbs the grating and causes the reflected light to have a wavelength shift which is a direct measure of the change in the strain or temperature.

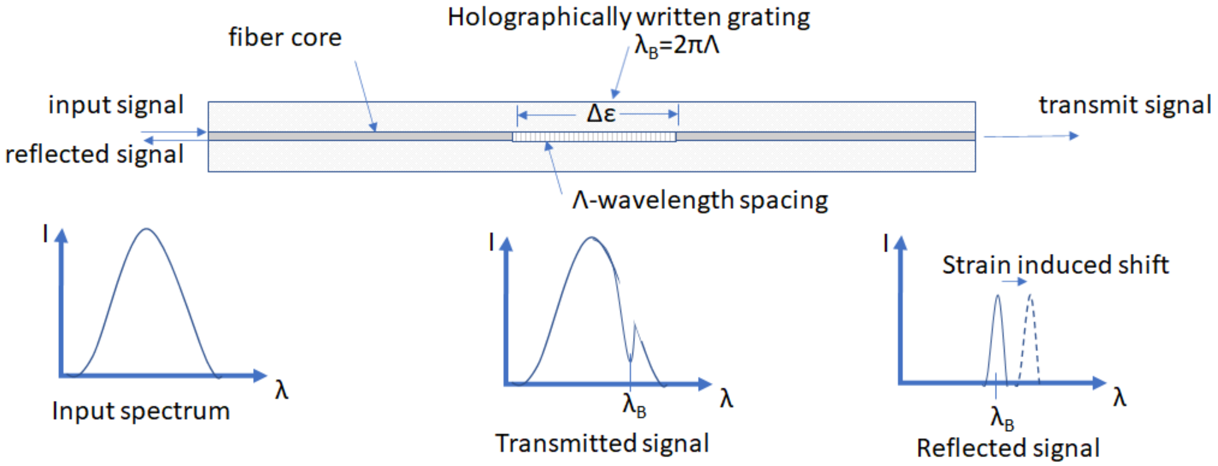


Figure 4.0 Wavelength Modulated Fiber Sensor for Light Detection

Figure 5.0 summarizes the common elements in a fiber sensing system design. Generally, all systems have a light source, interface to the optical fiber, a modulator that alters the light in a manner proportional to the perturbing environment and a photodetector. The options to achieve these functions are listed.

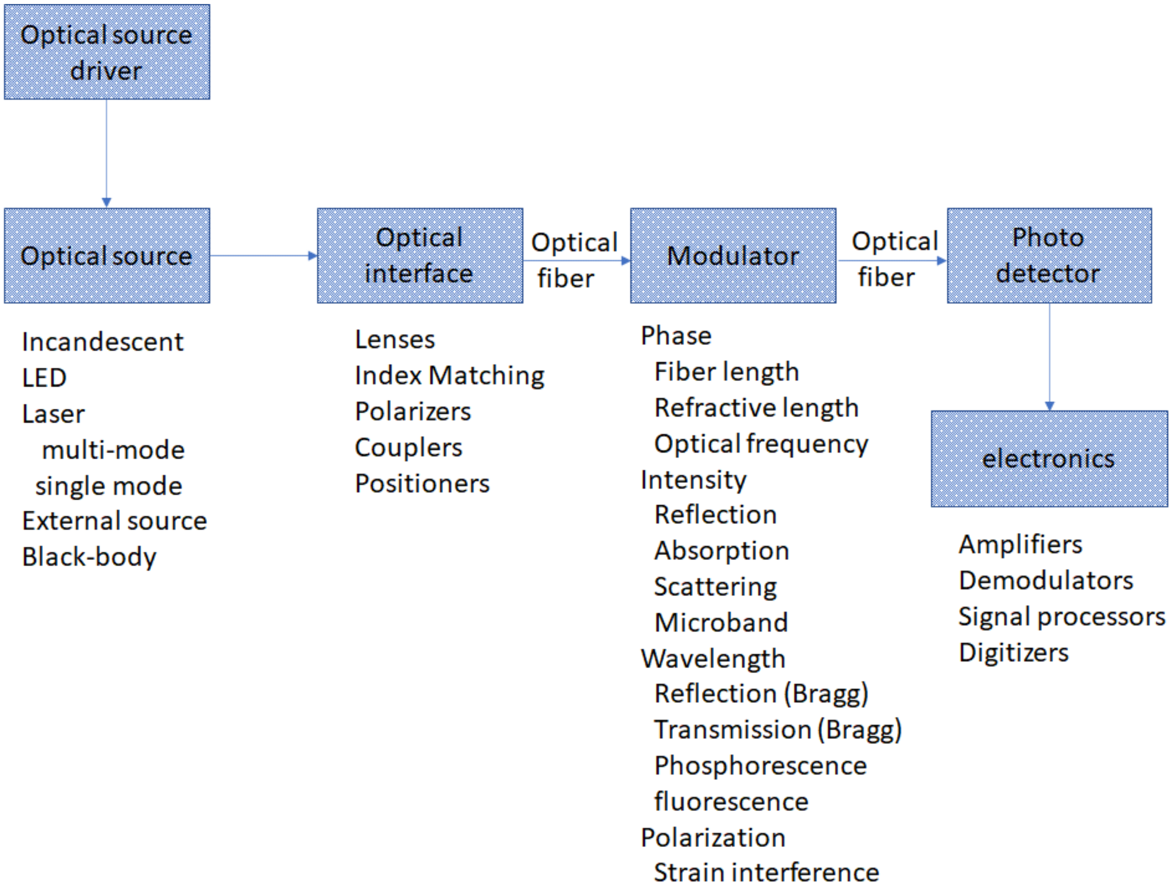


Figure 5.0 Common Elements in a Fiber Sensing System

3.0 Sensor Networks [6, 7]: Sensors monitor conditions within our homes, workplaces, vehicles, phones, and environment providing us with status information. The ability to combine data from several sensors and sensor types significantly increases the amount of information a sensor can provide; improving present status as well as often accumulating sufficient information to predict future conditions. Collecting measurement data from many sensors requires interconnecting sensor processors to communicate and exchange information. The sensor network provides the medium to implement the sensor interconnections. A sensor network connects many sensor systems, termed sensor nodes within the network, providing measurement data on quantities such as heat, pressure, and motion, as described in Part 1.0; transmitting the information to another node or central hub. A sensor node is the complete sensing system described in Part 1; sensor, sensor processor, power supply and output display and/or communication interface. Network design was pioneered by the Bell telephone system network. Patch panels gave way to switchboards that became automated evolving into switching centers as the network grew. Phone functionality and speed also grew and by the 1980s the wireless phones had been integrated into the network and were becoming common place. The digital revolution set up the groundwork for data transmission with routers functioning as switching centers for the Internet which is now the lifeline of the consumer and business industry. The network infrastructure for present day sensor networks can be wired, often using an ethernet interface, or wireless. With the rapid development of MEMS, sensor networks are a vital part of the Internet of things (IoT) as was illustrated in Figure 1.0. As sensors record measurements in many industrial and living environments, information collected can be continuously transmitted for processing and analysis over a network. With the sensor network connected to the Internet or other computer networks sophisticated processing can be utilized with processing tools and storage available via the Cloud. This is a major source of support for the network since one of the biggest issues with large quantities of sensor data measurements is processing and storage. Data can be used to enhance the fidelity of information and/or in real time sense and control a process or environment. The use of sensors connected via multiple sensor networks continues to grow; we use smartphones for nearly everything that involves communication and information exchange; traffic, directions, location via a global positioning system (GPS), and communication with each other. We have smart roads, parking, lighting, home environment monitoring and control, smart shopping, all made possible by sensor networking. Sensor networking is an old yet evolving technology; concepts remain valid but implementations will change as Smartphones grow in functionality relative to IoT and Cloud based applications.

The wireless sensor network (WSN) interconnects many nodes using wireless transmission. WSN are simple to install, reconfigure, and maintain and with the on-going development of sensor communication interfaces have become a key technology of the IoT. Radio frequency (RF) transmission is often used but there are distance limitations at the sensor node level due to sensor power and size constraints. Wireless sensor networks can interconnect spatially distributed autonomous sensors over wide areas; sensing the physical and environmental conditions of many locations, and cooperatively transmitting the information over the network to a main node. Without the need for cabling and associated constraints, the WSN can include a few or many nodes relying on reliable wireless connectivity and reconfigurability for growth and application. Transmission power constraints are compensated for using a cooperative network that relays data between sensor nodes. The relaying technique, termed multi-hop, uses methods providing a means to wirelessly transmit sensor data efficiently over long distances to the user. Each sensor node is responsible for; measuring a desired quantity; transmitting the information to an end user; as well as providing multi-hop support for other sensor nodes by relaying their data back to the end user. Multi-hop transmission techniques are similar to routing or in smaller networks broadcast transmission can be used. The communication exchange must use well defined processes or protocols for data exchange. The protocol defines a set of communication rules, formats, handshaking procedures, driver interfaces, and error recovery procedures implemented in both hardware and software. WSN are supported by many protocols at different communication layers but common design requirements are efficiency, speed, resource friendly, energy efficient, and secure.

As the number of networked sensors grows the need to minimize their size, weight, and power (SWaP) grows. Low weight, especially in mobile environments often goes go hand in hand with small size. SWaP constraints are common to many systems and the key challenge is to minimize SWaP without compromising sensor system (sensor, processor, communication I/O) performance or the potential for performance improvement. This can often be difficult, as sometimes mass coupled with size adds to the stability of sensor performance. In applications where an WSN communicates with an external network via a gateway, that provides access to the network, the possibility exists of using this bridge to share resources. If data can be stored and processed by devices with more resources, SWaP requirements on the local sensor network nodes can be reduced. Power limitations are a critical and challenging design issue for sensor nodes, especially if battery powered. Any sensor conditioning or processing, memory and the RF link transceiver components must all be low powered to conserve lifetime. Energy conversation as well as energy harvesting is an on-going area of WSN research. Scheduled duty cycling of energy consumption to maximize lifetime is a conservation method, nodes can be switched to a non-operating state extending lifetime, but this must be synchronized between nodes coordinating data collection requirements and minimizing network latency and routing overhead. Similarly, sensor nodes must report their power status and provide an estimated time remaining before they fall below an operating power level to allow for maintenance. Besides power efficiency, other characteristics of a WSN should include [6]:

- Robust sensor node and network configurations minimizing node failure impact
- Support any node mobility requirements
- Robust secure inter-node communication and control interfaces
- Network scalable and reconfigurable
- Environmentally hardened and user friendly

Many WSNs can be installed in different remote and hostile environments and adaptive network implementations are vital to sustaining good performance. Algorithms and protocols must address increased lifespan; robustness and fault tolerance; and self-configuration. Research is on-going to develop sensor network architectures that minimize routing information, routing traffic load, and energy consumption.

WSN use transmission technologies such as Bluetooth, Cellular, Wi-Fi or near field communication which could include fiber optic links to connect sensors. The IEEE 802.15.4

working group provides a standard for low power device connectivity, often used by sensors and smart meters for connectivity. The growth of IoT has spurned many other proposals that provide sensor connectivity. Some of the present wireless standards and solutions for sensor node connectivity in use include [6]:

- Thread and Zigbee can connect sensors operating at 2.4 GHz with a data rate of 250kbit/s [6].
- Z-wave @ 915 MHz and EU @ 868 MHz utilize the advantage of lower frequency for increased range but data rate is low at 50 kb/s. [6]
- Wi-Fi SoC, IEEE 802.11s; 2.4 GHz – 5.9 GHz; 50 Mb/s max
- Bluetooth
- Cellular
- LORA is a form of low power wide area network (LPWAN) providing long range low power wireless connectivity for devices, which has been used in smart meters [6].
- Narrowband IoT and long-term evolution for machines (LTE-M) can connect up to millions of sensors and devices using cellular technology [6].

Today, our lifestyles are tied to the smartphone or iPhone and it is reasonable to expect that these devices will take a more prominent role in WSN design. They are already used for point-to-point monitoring and control. Monitoring sensor nodes over such a device should be realizable and sending the information to a remote central node for analysis over a phone is not a major stretch and likely already being designed and used. Eventually some standard architecture will be developed for this implementation.

3.1 Sensor Network Topology [7]: The sensor network topology defines the interconnect geometry of the network sensor nodes. The sensor node includes the sensor, a power source, and a micro-computer for signal processing and communication. The computer interfaces to its local sensor collecting the data required and then transmits the information via a communication transceiver to other sensor nodes or a central node. There are several network topologies that can be used to arrange the sensor nodes. Baseline topologies include point to point, bus, star, tree, mesh, ring, circular and grid; all briefly described later in this section. There are also hybrid arrangements of these topologies with each topology having advantages and disadvantages. The application performance requirements and physical constraints will drive selection. In general; performance requirements will include data throughput to a central node, number of nodes, ease of maintenance and fault detection, network security and reliability and path redundancy to bypass a failed node. The network processing architecture usually tends to be either centralized or distributed, and sometimes a hybrid of the two. Figure 6.0 conceptually shows the components of a sensor field network connecting over the Internet to a central management node. The primary functional elements of a network are sensor nodes, gateways, and clients. Data collected at the sensor is transmitted via other nodes until it reaches a gateway node. It is transferred to a client management node providing a user interface to analyze and store the data. Sensor nodes serve the dual purpose discussed previously; sensing for data collection and as communication relay stations providing multi-hopping pathways for long range data transmission to a central node. The network topology design establishes the connectivity configurations between nodes that provide the optimal configuration for the application and to best support the multi-hop requirement. Since sensor node communication pathways must interact, with the advancements in processor technology it may be possible to distribute some of the data pre-processing among the nodes as opposed to only at one central node. The topic of distributed versus centralized processing will come up several times in this part of the course. The centralized network architecture has been used since the early days of networking, utilizing one central processing and control node. But this approach is also subject to single point failure. The reliability of the sensor network can be improved using a distributed control architecture.

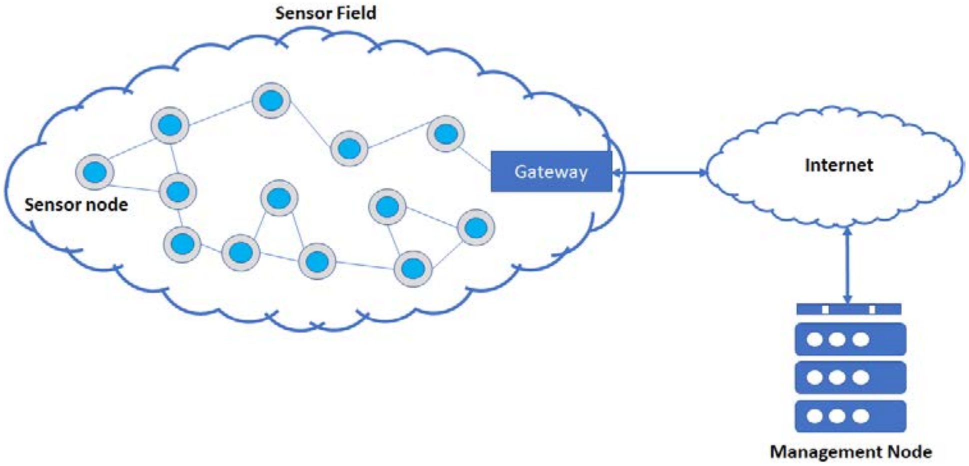


Figure 6.0 Concept Sensor Field Network Interface to Central Management Node

Distributed control can provide a broader allocation of resources; distributing data collection, processing, and storage among the network nodes based upon the status of the state of each node. This does not occur without some impact on the sensor node design, however. Distributed control can be complicated and may require a self-organizing processing algorithm, often based upon the network topology, and can adapt easier to a changing environment. However, this will require a higher end processor which could significantly impact sensor node SWaP. The numerical data collected from wireless sensor networks is usually stored in the management or central node. Network interoperability interfaces enable real time integration of different sensor networks with the Internet which can allow an authorized user the ability to monitor or control a WSN in a web browser. If nodes can analyze data received and transmitted, possibly performing some pre-fusion processing, such as identifying similar measurement readings from other sensor nodes which can minimize collection or transmission of redundant data as well as filtering out or eliminating data not required at all at the central node for processing. This can significantly reduce network data traffic. Correlating data from many spatially distributed nodes to identify similar measurements requires special data combinatorial techniques; but potentially reduces the amount of network traffic as well as sensor node energy consumption. However, while distributed processing at the sensor node level can reduce data traffic throughput, this sensor processing requires a higher end micro-computer and can add data latency to processing at the central node. Using network gateways to perform certain functions may alleviate some of these issues. Gateways can also be used to implement energy efficient scheduling algorithms for sensor node processing, improving network energy efficiency. There are several types of network topologies described below a few of which easily support a more distributed processing architecture while others a centralized architecture.

Point to Point Topology: In this simple topology, a secure communication path is provided by a single data communication channel with each device capable of operation as a client or a server.

Bus Topology [7]: In this topology, shown in Figure 7.0, one node sends data to another node on the network by broadcasting it on a central bus cable accessed by all nodes but only addressed to one recipient who accept and processes the message. Bus topology is easy to implement but latency results from transmission congestion on the single path communication. Limiting the number of nodes provides the best performance.

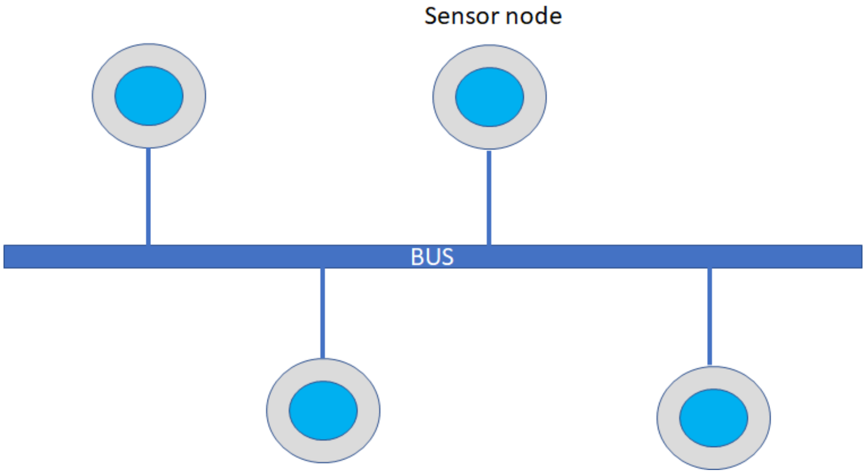


Figure 7.0 Bus Network Topology

Star Network Topology [7]: Every peripheral node in a star network is connected to a centralized communication hub (sink) acting as a server and the peripheral nodes clients. Each communication flows through a centralized hub and the nodes cannot communicate directly with each other. The entire communication must be routed through the centralized hub. It is easy to design and implement but does have the disadvantage that the hub can be a single point of failure, and congestion can occur if one node transmission is long. The topology is shown in Fig. 8.0.

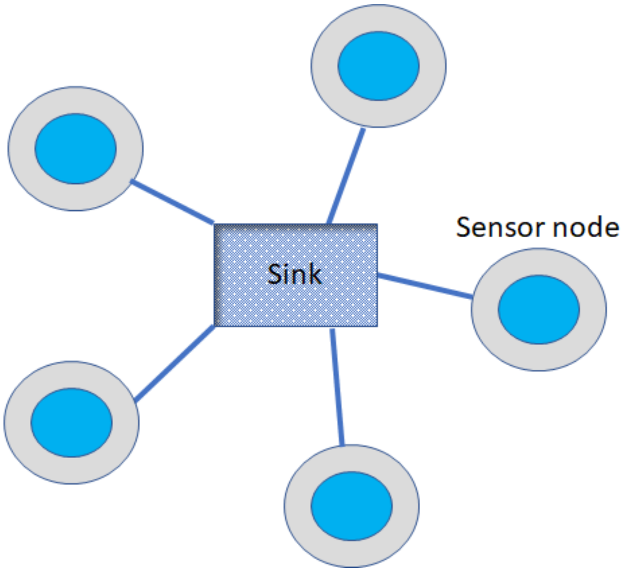


Figure 7.0 Star Network Topology

Tree Network Topology [7]: The tree network is effectively a collection of interconnected star networks. The tree network uses a central root hub as the main communication router to the central hub of each effective star network. In the hierarchy, the central hub is one level below the root node and routs messages between the peripheral nodes of its star network, these nodes being termed a leaf node. The tree network can be considered a hybrid of both the star and point to point networking topologies as shown in Fig 9.0.

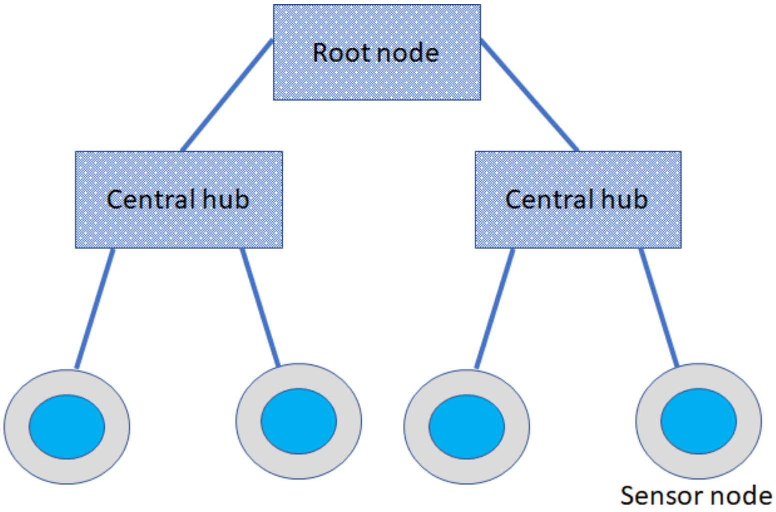


Figure 9.0 Tree Network Topology

Mesh Topology [7]: Mesh topologies allow data to flow along several paths from source to destination supporting multi-hop transmission. A mesh network in which all nodes are interconnected to each other is called a full mesh with the number of connections (C) required between nodes (N) being C=N*(N-1)/2. As connections and complexity increase quadratically, normally only small full mesh networks are implemented. A partial mesh networks is one in which some nodes connect only indirectly to others. This topology, full and partial, can support multi-hop transmission and is shown in Figure 10.0.

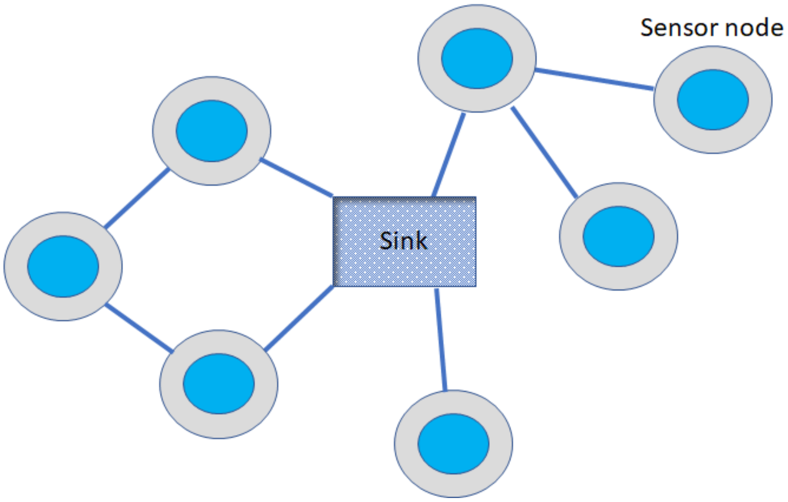


Figure 10.0 Mesh Network Topology

Ring Network Topology [7]: In a ring network, every node has exactly two neighbors for communication purposes. All messages travel through a ring in the same direction being repeated through each intermediate node until they get to the destination node. One node failure breaks the loop causing network failure but without the need for a server/client interface it reduces congestion of traffic and double path communication. The topology for this approach is shown in Figure 11.0.

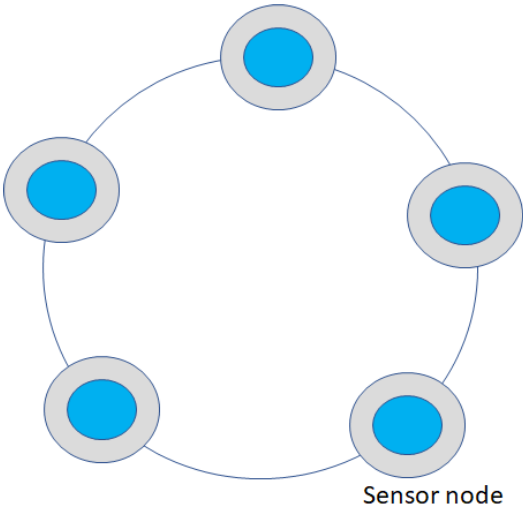


Figure 11.0 Ring Network Topology

Circular Network Topology [7]: This topology uses a circular sensing area centered about a sink node. Sensor nodes measure the quantity of interest, transmitting the data to the sink. The nodes are randomly distributed with uniform density in the circular sensing area around the sink, as shown in Figure. 12.0. Depending on a nodes signal transmission range and its distance to the sink node, data may have to be relayed through several nodes on a multiple hop path before it is received at the sink. The circular web topology is easy to establish, easy to maintain, and more efficient.

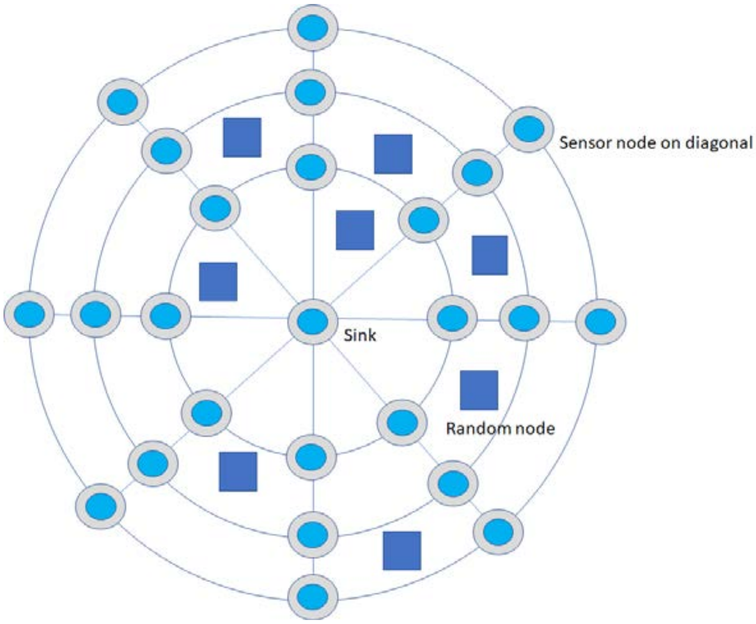


Figure 12.0 Circular Network Topology

Grid Topology [7]: The sensor network field is partitioned into squares as shown in Figure 13.0. Node operation within a grid can be sequenced to extend the network life time. Inside each grid, one node is selected as the master node (in red) to forward routing information and packet transmission with routing implemented between neighboring grids. Multi-path communication protocols route packets quickly and minimize congestion to ensure sensor node power is conserved.

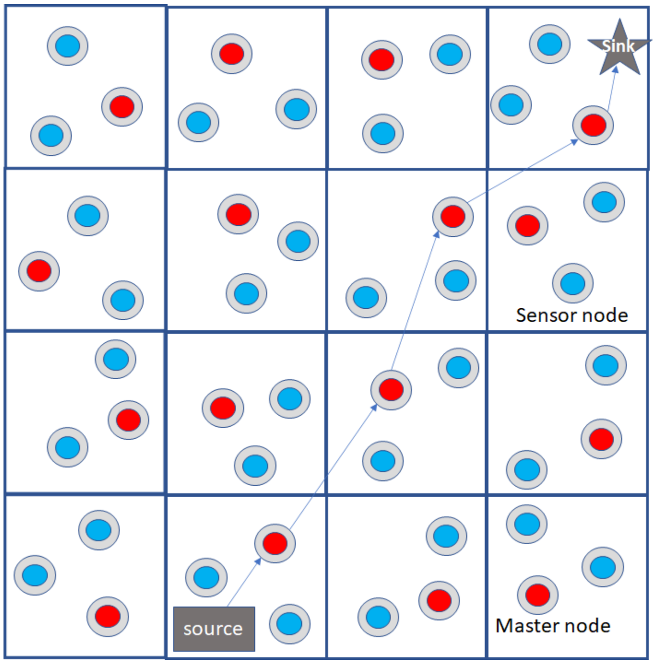


Figure 13.0 Grid Network Topology

The network topology defines the physical interconnect configuration of the sensor nodes. However, communication between nodes as well as with a central processing hub requires several functional procedures which are implemented using different protocols and algorithms. From the perspective of sensor network performance, having an understanding of these procedures is important as they can impact transmission latency as well as network control and recovery in the instance of a sensor and/or network failure.

3.2 Network Protocols [8, 9]: Communication between nodes uses protocols to exchange data. The protocol defines a set of procedures covering rules and formats for communication and communication interfaces. The protocols are defined within layered communication model structure, each layer defining a functional step for communication. Two telecommunication systems models used for protocol design are the open systems interconnection (OSI) model and the Internet protocol suite; also referred to as the transmission control protocol/internet protocol (TCP/IP) model. The OSI model uses an open architecture that define communication functional layers, but not the specific implementations in terms of algorithms or techniques. The rigid structure of the model architecture makes it less applicable to sensor networking but is worth a brief overview of the architecture since several layers are common with the TCP/IP model. The TCP/IP model defines communications protocols used by the Internet and similar computer networks with the baseline protocols being the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The TCP/IP model architecture is not as structured as the OSI model, providing more functional interfacing between layers that is more applicable to WSN. Both models are briefly described and for sensor networking there are many variants not described.

The Open Systems Interconnection model (**OSI model**) [8] uses an open architecture providing for a wide diversity of communication systems, each with their standard communication protocols, to perform the same communication functions within the model structure. Data flow is categorized by seven functional layers. The lowest level is the physical layer which for a sensor network would encompass the sensors and collection of sensor data. The five layers to transmit the data across a communication medium to the highest-level. The highest level uses the data in an application; such as the analysis and fusion of sensor data. Each layer has an interface to the layer above and below it, realized in software by standardized communication protocols. This inter-dependence between adjacent layers, while providing a good structure, limits interlayer interfaces between non-adjacent layers and potentially network performance. A brief description of each layer follows [8]:

Layer 1: The physical layer's function within the sensor network would be the interface to a sensor node; collecting data and transmitting the physical data to with a transmission medium. The physical layer represents the electrical and physical representation of the system, the physical sensor interface to a network.

Layer 2: The data link layer provides the communication link between two nodes connected in the physical layer for node-to-node data transfer. The protocol establishes and terminates a connection between nodes as well as providing for reliable data frame transmission and flow control between them. There are two sublayers:

- the Media Access Control (MAC) layer which provides access control for network devices and permissions to transmit, and
- the Logical Link Control (LLC) layer that identifies network layer protocols and performs error checking, and frame synchronization.

Layer 3: The network layer's function is to define the process for communication between different networks; the exchange of data packets between nodes connected in different networks given the node addresses. Most of the router functionality resides here and is responsible for packet forwarding, including transmission through different routers

Layer 4: The transport layer This layer coordinates data transfer between end systems and hosts; dealing with data format such as variable length data sequences, rate, and destination while maintaining the quality of service.

Layer 5: The session layer controls the connections between computers; establishing, managing and terminating the connections between the local and remote application. In the OSI model, when two devices, computers or servers need to communicate with one another, a session needs to be created and closed at the Session Layer.

Layer 6: The presentation layer maps network data formats used in the session layer to data formats required by an application. It provides independence from data representation, translating between application and network formats putting the data into a form that the application accepts.

Layer 7: The application layer is the OSI layer closest to the end user interacting directly with the software application such as sensor fusion. It receives information directly from users and displays incoming data to the user. This layer interacts with software applications.

The Internet Protocol Suite [9] specifies an end-to-end data communication process. Defining how data should be packetized, addressed, transmitted, routed, and received. Protocols used by the Internet are described by the TCP/IP model. TCP/IP defines four broad functional layers as determined by the operating scope of their protocols; from lowest to highest the layers are:

Layer 1: The link layer, the protocols of this layer operate based on the procedures used by the local network connection to which a host is attached. This layer includes all hosts accessible without traversing a router and corresponds to the OSI data link layer but may also include similar functions as the physical layer as well as some protocols of the OSI's network layer

Layer 2: The internet layer, this provides for Internet communication between independent networks, routing data from source network to a destination network supported an internet protocol (IP) addressing system providing host addressing and identification. It performs functions similar to those that are a subset of the OSI network layer

Layer 3: The transport layer, the host-to-host transport path; handling host-to-host local and remote network communication over routers; and maps to functions of the OSI session layer as well as transport layer

Layer 4: The application layer, the scope of the software application providing process-to-process data exchange for sensor fusion applications and maps to the OSI application layer, presentation layer, and most of the session layer.

The TCP/IP model does not define the presentation data format or specify the definition of the presentation and session layers between the application and transport layers as in the OSI model. Libraries and application program interfaces provide these functions. Transport layer protocols used in the application layer provide for a stable network I/O device connection and maintain transport layer connection information including IP addresses and port numbers.

The TCP/IP model is closest to that required for the WSN telecommunication model. Breaking out the link layer into a physical layer and a link layer would provide a typical 5-layer WSN model, or from lowest (physical) to highest (application) layer level [7]:

Layer 1: The physical layer addresses sensor node physical interfacing connections, connectors frequency selection, data encryption, transmission and receiving techniques.

Layer 2: The data link layer handles data sequence multiplexing and flow control, frame detection, media access control (MAC) and error detection and correction. The local network communication link control processes are specified as are the protocols used to describe the local network topology. The interfaces are defined to effect transmission of Internet layer data between neighboring nodes.

Layer 3: The Internetwork layer handles routing to the transport layer with multi-hop wireless routing protocols between sensor nodes and a gateway. It handles uniform networking and Internetworking interfaces within the network topology. IP addressing and routing structures used by the TCP/IP protocol suite are defined and functions similar to an IP router to transport data to the next node that is closer to the final data destination.

Layer 4: The transport layer maintains the sensor network data flow between lower layer levels and the network's application, establishing the path for communication with the sensor application. The user datagram protocol (UDP) is the transport layer protocol that provides low-latency reliable datagram transmission between Internet based sensor applications. It allows for data transfer enabling prior to handshaking with the Transmission Control Protocol (TCP) providing reliable flow-control.

Layer 5: Application layer handles different types of sensor fusion application software used for the various sensing tasks creating data for analysis and control.

A simple transport layer topology, shown in Figure 14.0, interconnects two hosts by way of a link between each host router. Each host application software executes read and write commands as if these functions were directly connected to each other.

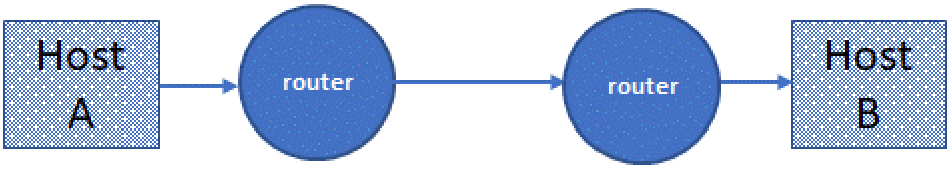


Figure 14.0 Simple Network topology [9]

Once the connection in established, lower layer communication protocols and algorithms define the interface so that communication details are transparent to the application. Communication at the transport layer is basically host-to-host, application data structures and the connecting router information is transparent to the connection. At the Internetwork layer, network interfaces are crossed at the routers which provide a bridge between networks. This data flow is conceptualized by the block diagram illustrated in Figure 15.0.

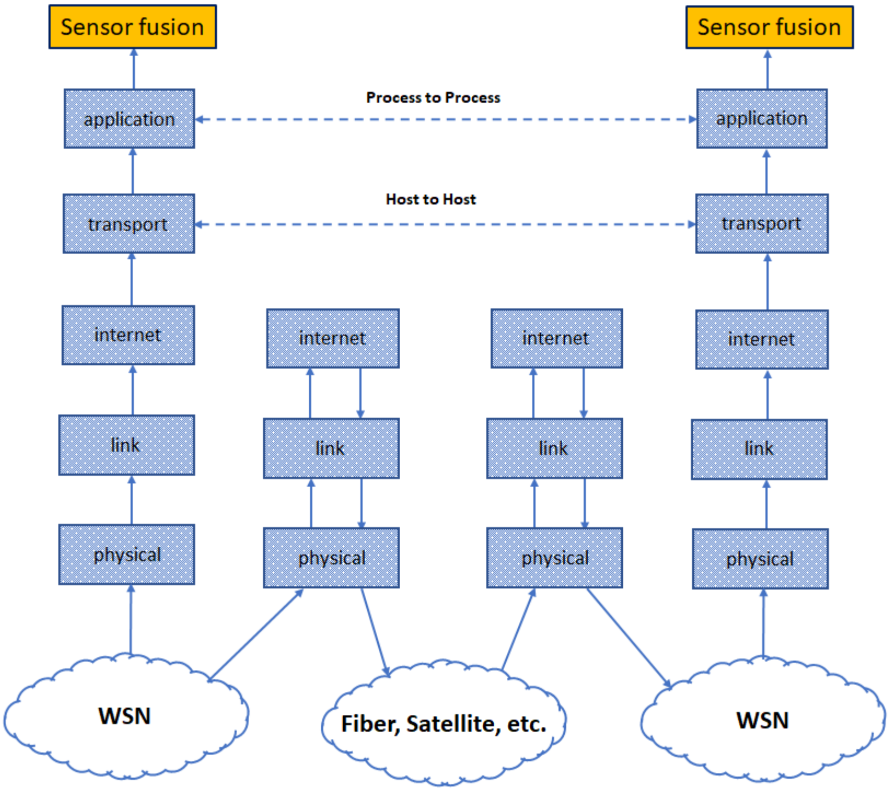


Figure 15.0 Communication Data Flow [9]

One final diagram, Figure 16.0, for this section shows the path of sensor measurement data coming over the link and flowing up through the layers to the application layer for sensor fusion. Sensor data is first converted for Internet transmission and then to a user datagram protocol (UDP) for transport to the application.

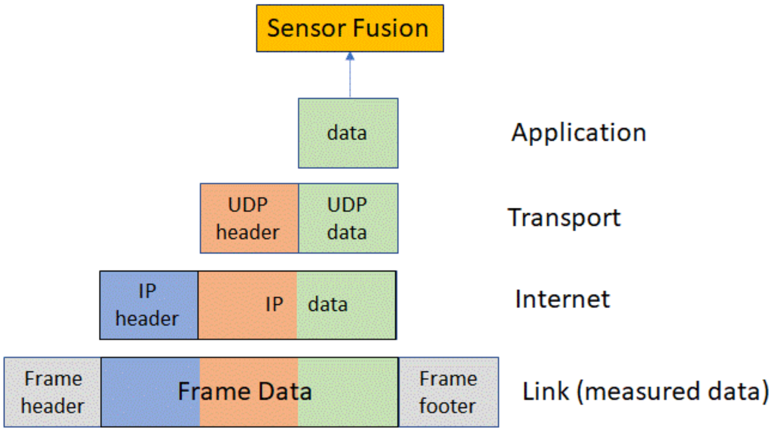


Figure 16.0 Application Data Flow Down Through Layers [10]

Sensor networks need to be reconfigurable and adaptive, as well as deal with some unique operating characteristics such as power conservation. Specific layering protocols may not always provide the best performance. The ability to cross over layers can often improve sensor network performance. An algorithm termed cross-layer optimization addresses limitation with the layered approach, effectively establishing a path between the layers required to improve performance and QoS (Quality of Service). Cross-layer functions are not attached to a specific layer and can affect more than one layer. Network management and security crosses most layers as does power management for efficient power load control of a sensor node. Similarly, sensor node connectivity configuration management and sensor node task management effect several layers. Cross-layer functions bridge disconnects with the functional layering approach improving confidentiality, reliability, availability, lifetime, and performance of the nodes and network.

3.3 Sensor Software [6, 11, 12, 13]: Software requirements are highly dependent on the amount of sensor node processing required. With a centralized processing node architecture, WSN operating systems can be less complex when used for the single function of sensing and raw data collection/transmission. With a distributed architecture, more processing is required at a node requiring a higher end processor. Power constraints can still be a limitation. Many small processor and processing configurations exist that include mini-processors, microcontrollers, DSP and FPGA. Arduino microcontrollers are designed to implement a single function while the Raspberry Pi is closer to a mini-processor having the potential for more processing functionality. WSN typically address a specific application, while maintaining the goal of a sensor node that is low cost and power, WSN development favors low-power microcontrollers. Some of the available microcontrollers can support several different real time programming languages; many C or C++ based. Other embedded operating systems include:

- uC/OS [11] can be used for sensor networks. uC/OS is a real time operating system (RTOS) written in ‘C’. Several functions can be defined and then executed as independent threads or tasks. Tasks are executed based on priority, lower priority tasks can be preempted by higher ones, and each task runs independently, like it had its own central processing unit (CPU). OS services include task management, memory, communication, and timing.
- TinyOS [12] is an operating system designed specifically for wireless sensor networks. It is a language that is event driven composed of event handlers and tasks programmed to run-to-completion. An external event, such as an incoming data packet or a sensor reading, triggers the TinyOS to signal the proper event handler to schedule tasks within the TinyOS kernel.
- LiteOS [13] is also an OS designed for wireless sensor networks, providing UNIX-like constructs and support for the C programming language. It is an open-source OS for IoT smart terminals supporting several different microcontrollers architectures (ARM, x86, RISC-V).

Other languages being used on mini-processors are Python and Java. Data storage for large networks can often be a challenge. Collaborative sensor data management [6] services can be obtained online that provide database services to WSN developers; connect to an online database for storage. These systems may slowly be replaced or integrated with the Cloud, providing developers connectivity to highly diverse databases with the ability to design their own applications.

4.0 Sensor Fusion [14]: The previous section discussed the formation of sensor networks and methods for transmitting sensor data to a central node or operations center. This section discusses sensor fusion algorithms that process the measurement data collected from many networked sensors and sensor types to obtain the best estimate of a physical quantities state and possibly even make decisions on how to control it. Using a network of sensors and sensor types can potentially result in greater accuracy and fidelity in a state estimate than obtained from a discrete sensor due to the additional information, as well as correlating measurements to estimate sensor performance, calibration, and alignment. More sensor information increases the amount of information on what is affecting a quantity's state; enhancing the understanding of the full state. Sensor fusion has been part of many complex systems for years. Robotics, airborne mounted optical sensors, cameras, LIDAR, and medical centers are just a few of the applications that use multi-sensor designs to achieve a desired performance. Healthcare facilities, medical centers and medical electronics represent applications where the need for sensor fusion is tantamount. If one considers a medical center; there must be a building environmental sensing system, a security sensing system, and the patient vital signs sensing system and database. Accelerometers and gyroscopes are integrated into wearable sensor systems used in a variety of medical environments such as physical therapy, charting the motion of legs and arms to determine if exercises are preformed properly. In general, wearable sensors; blood pressure, heart rate, temperature, etc. could send data to a common medical database, possibly via a smart phone, where it is fused to track wellness and any developing medical condition, with telehealth services providing status and alerts accessible by the family physician. This information database and fusion analysis may even be part of the cloud allowing it to be accessed and reviewed by doctors at anytime and anywhere. Sensor fusion can be used for numerous other applications which includes feature assessment associated with object detection, recognition, identification for situational awareness and object tracking. With sensor applications growing at a fast pace, commensurate with technology advancements, more focus on sensor networking and fusion is required. Sensor fusion is evolving in our interconnected environment to meet the sensing system design challenges presented by ever-growing sensor networks, to extract more information with greater accuracy from the large sets of noisy sensor measurements under all sensor operating conditions. On-going sensor technology and processing hardware and software developments support the need for real-time data fusion of large sensor networks with multiple combinations of wireless MEMS sensors.

To meet the many challenges, sensor data fusion algorithms use advanced processing techniques that include artificial intelligence (AI), neural networks, probabilistic decision theory and networks, digital signal processing, and control theory. Learning algorithms can be used to improve accuracy and fusion processing time. The self-driving car is an example of an application that relies heavily on sensor fusion. The algorithms use sensor data from multiple radars, LIDARs and cameras to generate a situational awareness model of the environment surrounding a vehicle. All the additional information these sensors provide far exceeds that which could be obtained from a single sensor and the processed fusion information can be transmitted to neighboring vehicles for increased safety and reduction in congestion. Whether or not autonomous vehicles enter the mainstream of our transportation system is to be seen; but its use in unmanned air, ground, and underwater vehicles is imminent. With an array of different sensors, each sensor type will have conditions for which its measurement technology works well and those for which it does not. Sensor fusion brings together the data from each sensor type using software algorithms to provide the best situational awareness model of the present environment possible with improved measurement accuracy and fidelity. The ultimate goal of sensor fusion is to emulate the human body, using the brain as a processor to fuse data from the nervous system and visual, audio, smell and taste sensing inputs allowing us to perform incredibly complex tasks. Typical sensors used for data fusion were described earlier and in Part 1 of the course.

4.1 Sensor Fusion Algorithms [14]: Sensor fusion is implemented using many methods and algorithms, including:

Central Limit Theorem [15]: The central limit theorem (CLT) establishes conditions for summing independent random variables such that their properly normalized sum can be described by a normal distribution even if the original variables are not normally distributed. This allows for probabilistic methods used for normal distributions being applied to problems that involve other distributions. The central limit theorem states that the distribution function for a sum of independent and identically distributed random variables with finite variances approaches that of a normal distribution function as the number of variables increases. This could also be stated within the context of the law of large numbers; as the sample size of a measured quantity increases, the average value of those samples will tend towards a normal distribution. This relates to sensor fusion because it says the more sensor measurements of a quantity obtained, the closer the distribution of the measurement sample averages will resemble a normal curve and approach the set's true average value. The more accurate a measured average value is, the less noise will factor into sensor fusion algorithms.

Kalman Filter: The Kalman Filter algorithm was briefly described in Part 1 of the course. It is an algorithm that uses a sequence of noisy measurements observed over time to obtain a more accurate state estimate than one based only on a single measurement by estimating a joint probability distribution over the variables for each timeframe. As mentioned in Part 1, it is based upon substantial mathematical statistical theory but in its simplest sense can be considered an error correction model. A measurement is compared to a predicted measurement; this error is used to correct the estimated state and then the corrected state inserted into the state propagation model to determine the state update for the next time step. The updated state is used to predict the next measurement. The error correction model multiples the error by a gain; known as the Kalman gain which is the focus of filter design and provides an optimum correction under certain conditions.

Bayesian Networks [16]: A Bayesian decision network is a probabilistic network model for a set of variables and their conditional dependencies. Bayesian networks are excellent for predicting the likelihood that several possible causes are contributing factors of an event that has occurred; such as the probabilistic relationships between diseases and symptoms. Given symptoms, the network would determine the probabilities of various diseases being the cause. Bayes' rule, based on joint probability, provides a foundation for the probabilistic models and network used to predict the likelihood that any one of several hypotheses is the contributing factor in a given event. As they are a powerful tool, they are described in more detail in section 4.3 of this course.

Dempster-Shafer [17]: This algorithm uses the theory of belief functions to provide a general framework for reasoning with uncertainty. It connects to other mathematical stochastic frameworks such as probability, possibility and imprecise probability theories. First proposed as a technique for evaluating statistical inference, the theory evolved into a general methodology for the mathematical modeling theory of evidence; allowing evidence from different sources to be combined to determine the degree of belief, using a belief function, that is consistent with all the available evidence.

Convolutional Neural Network (CNN) [18]: The convolutional neural network (CNN) is also termed shift invariant neural networks because of the architecture for assigning weights and translation invariance characteristics. Their origins are in biological models of the human brain, therefore use some common terminology, and have applications that include image and video recognition, image classification, and medical image analysis. The basic working unit of the CNN is the neuron which is also a vital network element communicating with other neurons. For image analysis, CNNs are used to pull out useful features from the image. Several layers of neurons are used with each gathering data with a specific level of detail, for example pixels, lines, shapes, and finally objects. This is followed by a classifier network that classifies the image as useful or not depending on performance objectives. The CNNs property of translational invariance refers to its ability to learn a specific feature or object and correctly classify it regardless of orientation or position in the image. In machine learning, the algorithms perform supervised learning of binary classifiers; the neuron hosting the processing algorithms and also communicating with other neurons. CNNs are fully connected networks such that each neuron in one layer is connected to all neurons in the next layer. CNNs technique to achieve regularization (avoid overfitting to smooth data) is by taking advantage of hierarchical data patterns and splicing together more complex patterns using smaller and simpler patterns. CNN based methods simultaneously process many channels of sensor data, producing classification results from the fused data based on feature recognition.

Artificial Intelligence [19]: Artificial intelligence is method applicable to sensor fusion to extract sensor information that can determine a quantities behavioral patterns, deterministic or stochastic. AI algorithms analyze the environment and respond to improve the likelihood of successfully achieving an objective as defined by a utility function. AI provides a system that can interpret and learn from external data, and adapt it to achieve specific goals and tasks. AI algorithms improve performance by developing new learning strategies or using old ones that have worked well previously. Learning algorithms can be based on Bayesian networks, decision trees, or nearest-neighbor. With enough data, they can approximate nearly any function. AI applications can use several analytical techniques including; formal logic, Bayesian inference, nearest-neighbor and even the more complex neural networks, described in the last paragraph. Learning algorithms work on the basis of past performance. Strategies, algorithms, and inferences that were successful in the past are likely to be in the future.

Three simple examples of sensor fusion calculations to improve measurement accuracy [14], based upon the CLT, a Kalman Filter, and a Bayesian estimator follow. The examples combine information from two sensors. Let X_1 and X_2 denote two sensor measurements with noise variances σ_1 and σ_2. A combined measurement X_3 can be obtained using the Central Limit Theorem as follows. For random variable with a normal distribution, the integrand is defined by the EXP(-X/σ^2). The sum of two normal random variables would be EXP(-[$X_1/\sigma_1^2 + X_2/\sigma_2^2$]). It is reasonable to define a combined measurement as [14]:

$$\frac{X_3}{\sigma_3^2} = \frac{X_1}{\sigma_1^2} + \frac{X_2}{\sigma_2^2}$$

or

$$X_3 = \sigma_3^2 \cdot (\sigma_1^{-2} \cdot X_1 + \sigma_2^{-2} \cdot X_2)$$

where

$$\frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

therefore

$$\sigma_3^2 = \frac{\sigma_1^2 \cdot \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

where σ_3 is the variance of the combined estimate. It can be seen that the fused result is simply a linear combination of the two measurements weighted by their respective noise variances. Measurements that have a smaller noise variance will dominate the combined value.

Another method to fuse two measurements is to use the optimal Kalman filter, described in Part 1 of the course. Assume the data is generated by a first-order system and let P_k denote the error covariance solution of the filter's Riccati equation (error covariance extrapolation). Applying Cramer's rule within the gain calculation, the filter gain is given by [14]:

$$L_k = \left[\frac{\sigma_2^2 \cdot P_k}{(\sigma_1^2 + \sigma_2^2) \cdot P_k + \sigma_1^2 \cdot \sigma_2^2} \quad \frac{\sigma_1^2 \cdot P_k}{(\sigma_1^2 + \sigma_2^2) \cdot P_k + \sigma_1^2 \cdot \sigma_2^2} \right]$$

and the combined measurement is:

$$X_3 = \frac{\sigma_2^2 \cdot P_k \cdot X_1}{(\sigma_1^2 + \sigma_2^2) \cdot P_k + \sigma_1^2 \cdot \sigma_2^2} + \frac{\sigma_1^2 \cdot P_k \cdot X_2}{(\sigma_1^2 + \sigma_2^2) \cdot P_k + \sigma_1^2 \cdot \sigma_2^2}$$

It is observed that when the first measurement is noise free, the filter ignores the second measurement and vice versa. The combined estimate is weighted by the quality of the measurements.

An estimator that can provide the best mean estimate of n-measurements is the Bayesian estimator. This estimator provides a best estimate of the mean of the measured quantity based upon prior information on the variance of the mean, β, and a best guess of the prior mean value, ν. Assuming the measured data is normally distributed N(μ, σ^2} and the mean is normally distributed N(ν, β^2) the joint density of the data and μ is given by [20, 21]:

$$\frac{\exp\left(\frac{-\sum_n (x_i - \mu)^2}{2 \cdot \sigma^2}\right)}{(2 \cdot \pi)^{\frac{n}{2}} \cdot \sigma^n} \times \frac{\exp\left(\frac{-(\mu - \upsilon)^2}{2 \cdot \beta^2}\right)}{(2 \cdot \pi)^{\frac{1}{2}} \cdot \beta}$$

Expanding and summing the exponents, the best posterior estimate of a mean value is obtained as [21]:

$$\bar{\mu} = \frac{\frac{\bar{x}}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\beta^2}} + \frac{\frac{\upsilon}{\beta^2}}{\frac{n}{\sigma^2} + \frac{1}{\beta^2}}$$

where

$$\bar{x} = \frac{\sum_n x_i}{n}$$

4.2 Sensor Fusion Processing [14]:

At a functional level, fusion can be categorized in terms of data conditioning, resolving feature characteristics in the data, and using the data and features to make a decision to classify and take an action.

- If the objective is be to improve the measurement accuracy by conditioning and fusing measurements from several sensors of the same type, this would be referred to as data fusion. Sensors work in a redundant configuration; each node delivers independent measures of the same quantity property; providing measurement error correction by comparing information from multiple nodes. In general data level fusion algorithms combine multiple sensor measurements to obtain information more accurate than that of a single sensor measurement. Some simple example algorithms were provided in the previous section.
- The evaluation or status of a complex quantity may require measurements of several quantities with different types of sensors. The different quantities necessary to evaluate the condition of the complex quantity are often referred to as feature characteristics or simply features derived from the measured data. Feature-level fusion uses measurements from different sensor types to characterize features which are combined into a feature vector for evaluation of the complex quantity. Sensors work in a complementary or cooperative fashion, the sensor configuration using many different measurements to obtain information to fuse the measured data into a feature vector used by decision-making algorithms. Motion recognition applications often use complementary features with algorithms such as a Bayesian network. The feature vector may be determined at the sensor node level which would decrease the data processing load as well as the amount of data sent to a central node. However, it increases processing requirements and thereby SWaP at the sensor node, also making timing difficult to control; requiring a very accurate time tagging system.
- A final level of fusion is termed decision-level fusion which results in an action or status evaluation of a quantity. This may be the result of a feature vector assessment or using measurement information fused from both data and feature level fusion. A set of hypotheses are determined from feature processing information provided by the individual sensor nodes. Features from several different types of sensors are fused together and combined with decision hypothesis testing providing a complete situational awareness map of the environment measured by the sensor network

When processing data from large sensor fields, especially if using portable devices, there are often limits on memory and transmission bandwidth. These constraints may require using data compression techniques to conserve data memory storage requirements as well as reduce the required communication bandwidth since less data is transmitted.

This topic of centralized versus distributed architectures has already been discussed under network topology but also impacts the algorithms used for sensor fusion. In general, it will depend on the application, performance required, and system constraints. In a centralized system, data is forwarded to a central node for sensor fusion. In a distributed system, each sensor node performs some sensor fusion processing as described previously. Combinations of centralized and distributed systems are also used to take advantage of the best characteristics of both architectures. Performing some processing at the sensor node can reduce the complexity of coordinating data transmission between sensors making scheduling and route planning easier. But this also requires time tagging and alignment of the data sent to the central node. There are always systems latencies and these must be accounted when collecting sensor measurements.

4.3 Sensor Fusion Techniques and Applications:

An example of using the Bayesian network is provided along with examples of other sensor fusion applications with for an inertial navigation system and an autonomous vehicle.

Bayesian Network [16, 20]: Bayesian statistical theory; estimation, inference, and the Bayesian network are the foundation of many sensor fusion algorithms. They are described further in this section due to their importance. Bayesian networks provide a graphical node configuration with each node representing random variables or quantities that can be quantified statistically with a probability function whose input is dependent on variables from other nodes, termed parent variables. The node output is the conditional probability of an event or occurrence calculated from the conditional probability function given the condition of the parent nodes. The key theoretical basis for network interconnections and node calculations is Bayes theorem. Based on joint probability that is a function of two variables A and B the joint probability can be expressed as the product of the conditional probability of either value A relative to B or B relative to A and the probability of the variable the conditional probability is relative to. Symbolically this can be expressed as:

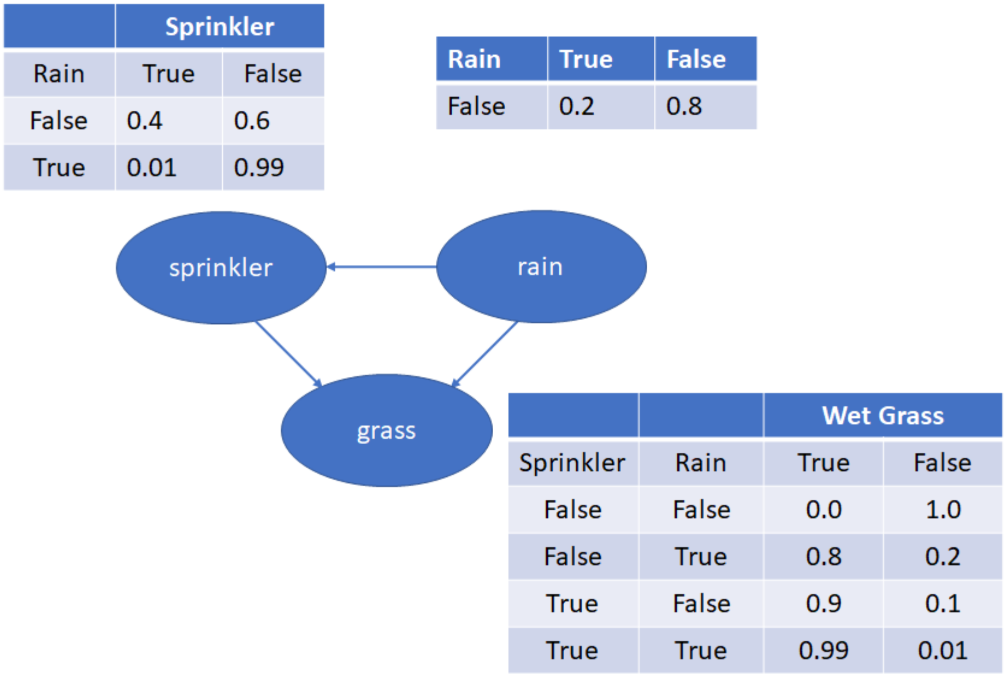
$$P(A, B) = P(A|B)*P(B) = P(B|A)*P(A)$$

or $$P(A|B)*P(B) = P(B|A)*P(A)$$

therefore $$P(A|B) = P(B|A)*P(A)/P(B)$$

This last expression is the key result and can be applied to n variables. The probabilities P(A) and P(B) are termed the prior probabilities of A, B and the conditional probabilities the posterior probabilities. A simple example, taken from [16], is provided to illustrate the concept and structure. A grass lawn (G) is integrated with a sprinkling (S) system. We want to determine the probability of the grass being wet due to rain (R). It is assumed that when it rains the sprinkler is shut off. The Bayesian network model with associated conditional probability tables (CPT) is shown in Figure 17.0 and we are interested in the probability of rain given the grass is wet or:

$$P(R=T|G=T)=\frac{\mathrm{P}(G=T,R=T)}{\mathrm{P}(G=T)}=\frac{\sum_{S\in T,F}\mathrm{P}\,(G=T,S,R=T)}{\sum_{S,R\in T,F}\mathrm{P}\,(G=T,S,R)}$$



	Sprinkler	
Rain	True	False
False	0.4	0.6
True	0.01	0.99

Rain	True	False
False	0.2	0.8

		Wet Grass	
Sprinkler	Rain	True	False
False	False	0.0	1.0
False	True	0.8	0.2
True	False	0.9	0.1
True	True	0.99	0.01

Figure17.0 Bayesian Network Model

The last term is the ratio of the joint probability function summed over all values of S, R. The numerator term is summed over values of S (i.e., S=0/1) while the denominator is summed over all values of S, R. The numerator term is given by:

$$\sum_{S\in T,F}\mathrm{P}(G=T,S,R=T)=\mathrm{P(G|S,R)\cdot P(S|R)\cdot P(R)+P(G|\bar{S},R)\cdot P(\bar{S}|R)\cdot P(R)}$$

This expression can be evaluated using the CPT in Figure 17.0.

$$\sum_{S\in T,F}\mathrm{P}(G=T,S,R=T)=0.99\cdot 0.01\cdot 0.2+0.8\cdot 0.99\cdot 0.2=0.1604$$

The expression for the denominator is obtained as:

$$\sum_{S,R\in T,F}\mathrm{P}(G=T,S,R)=\mathrm{P(G|S,R)\cdot P(S|R)\cdot P(R)+P(G|\bar{S},R)\cdot P(\bar{S}|R)\cdot P(R)+}$$
$$\mathrm{P(G|S,\bar{R})\cdot P(S|\bar{R})\cdot P(\bar{R})+P(G|\bar{S},\bar{R})\cdot P(\bar{S}|\bar{R})\cdot P(\bar{R})}$$

This expression is evaluated again using the CPT in Figure 17.0.

$\sum_{S,R\in T,F} \mathrm{P}(G = T, S, R) = 0.99 \cdot 0.01 \cdot 0.2 + 0.8 \cdot 0.99 \cdot 0.2 + 0.9 \cdot 0.4 \cdot 0.8 + 0.0 \cdot 0.6 \cdot 0.8 = 0.1604 + 0.288 = 0.4484$

The ratio of the two joint probabilities can be evaluated providing the probability the grass in wet due to rain.

$$\frac{\sum_{S\in T,F} \mathrm{P}\ (G = T, S, R = T)}{\sum_{S,R\in T,F} \mathrm{P}\ (G = T, S, R)} = \frac{0.1604}{0.4484} = 0.3577 \rightarrow 35.77\%$$

For this example, rain is the main parent variable to both the sprinkler state and the grass state. The sprinkler and rain are both parents to the grass state.

Bayesian network theory identifies state variables that define an all-inclusive model to evaluate the probability for a condition. Updates are made to the state variables as other variables, termed evidence variables, provide further sensor measurement information related to the condition. Computing the conditional or posterior distribution of variables given evidence (sensor information) is called probabilistic inference. The posterior distribution provides statistics sufficient for detection and decision-making applications [16]. Bayes' theorem is applied, within the context of a Bayesian network, to highly complex problems with the joint probability density function for a Bayesian network expressed as a product of the individual density functions, conditioned on their parent variable [16] as:

$$p(x) = \prod_{v\in V} p(x_v | x_{pa(v)})$$

where pa(*v*) denotes the set of *v* parent nodes. Given a set of random variables, using this definition the joint probability distribution is calculated from conditional probabilities using the chain rule as [16]:

$$P(X_1 = x_1 \dots X_n = x_n\ ) = \prod_{v=1}^{n} P(X_v = x_v) | X_{v+1} = x_{v+1} \dots X_n = x_n)$$

From the definition above, given values for the parent variables and using the conditional independence of the non-parent variables, this can be expressed as [16]:

$$P(X_1 = x_1 \dots X_n = x_n\ ) = \prod_{v=1}^{n} P(X_v = x_v) | X_j = x_j for\ each\ X_j\ that\ is\ a\ parent\ X_v)$$

Room Access and Presence Security [22]: The use of the Bayesian network is illustrated with one final example applied to securing and monitoring the entry, presence and exit status within a closed facility. The distributed sensor network makes use of five sensor types: WSN, radio frequency identification (RFID) readers, door sensors, sensors on actuators, and software sensors. WSN is a network of small computing devices interfacing to off-the-shelf sensors for measuring ambient quantities and with wireless transceivers that enable data exchange with neighboring nodes. Sensor nodes installed in rooms were close to sensitive indoor areas: by the door, window, and the user's desk. User presence was implemented using an audio sensor capable of detecting noise levels in its proximity, providing an indication of room occupancy. The RFID read sensor nodes, installed close to the main entrance and office doors, read RFID tags imbedded in personnel badges enabling monitoring and control of access. Each entrance had a sensor to monitor door status providing open/close/locked information back to the sensor network. The on/off status of equipment such as air conditioning and room lighting is also monitored. Detecting interaction between these sensors provides a reliable indication about the presence of at least one person in the monitored area. Finally, computer software acts as a sensor determining the status of a person's computer activity, logged on/off and terminal use. The architecture is shown in Figure 18.0. There are three functional blocks, the top block is the adaptive optimization algorithm that use algorithms providing network and sensor reconfiguration based on sensor performance, level of confidence they afford the Bayesian network, and associated cost to the sensor network. These algorithms layer effectively allows sensors to be activated or deactivated. The Bayesian network in the middle performs the sensor fusion of the data from the sensor network at the bottom. The Bayesian network estimates the probability of occurrence of a state feature given a set of input sensor data measurements. Within the context of this example, sensor nodes are defined as evidence nodes with E^i, being an estimate of that state status. The relationship between the current state and sensed status is provided by a probabilistic sensor model $P(E^i_t |X_t)$. The current state depends on the past state according to a Markov state transition probability $P(X_t|X_{t-1})$. In general, the belief in the specific value of a state variable is the conditional probability with respect to the all observations from the initial time to the current time. A belief function is defined as [22]:

$$Bel(x_t) = P(x_t|e_1^1, e_1^2, \ldots e_1^n \;\ldots\ldots e_t^1\,, e_t^2\,, \ldots e_t^n) = P(x_t|E_1, E_2, \ldots E_t\;) = P(x_t|E_{1:t})$$

Given the Markov assumption and the conditional independence of the sensor measurements, once the state induced by the Bayesian network is known, Bayes theorem is used to determine the belief about the current state as [22]:

$$Bel(x_t) = \eta \cdot \prod_{e_t^i} P(e_t^i|x_t) \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}) \cdot Bel(x_{t-1})$$

Due to the Markov assumption and conditional independence of measurements, at each time interval only a reduced set of variables is required significantly decreasing the computational load. The adaptive optimization layer implements an algorithm that can reconfigure the sensor network. Modifications may be necessary due to particular environmental conditions, or poor performance from the present system. This is accomplished by activating or deactivating sensors or modifying the information flow to the probabilistic inference algorithm to improve performance.

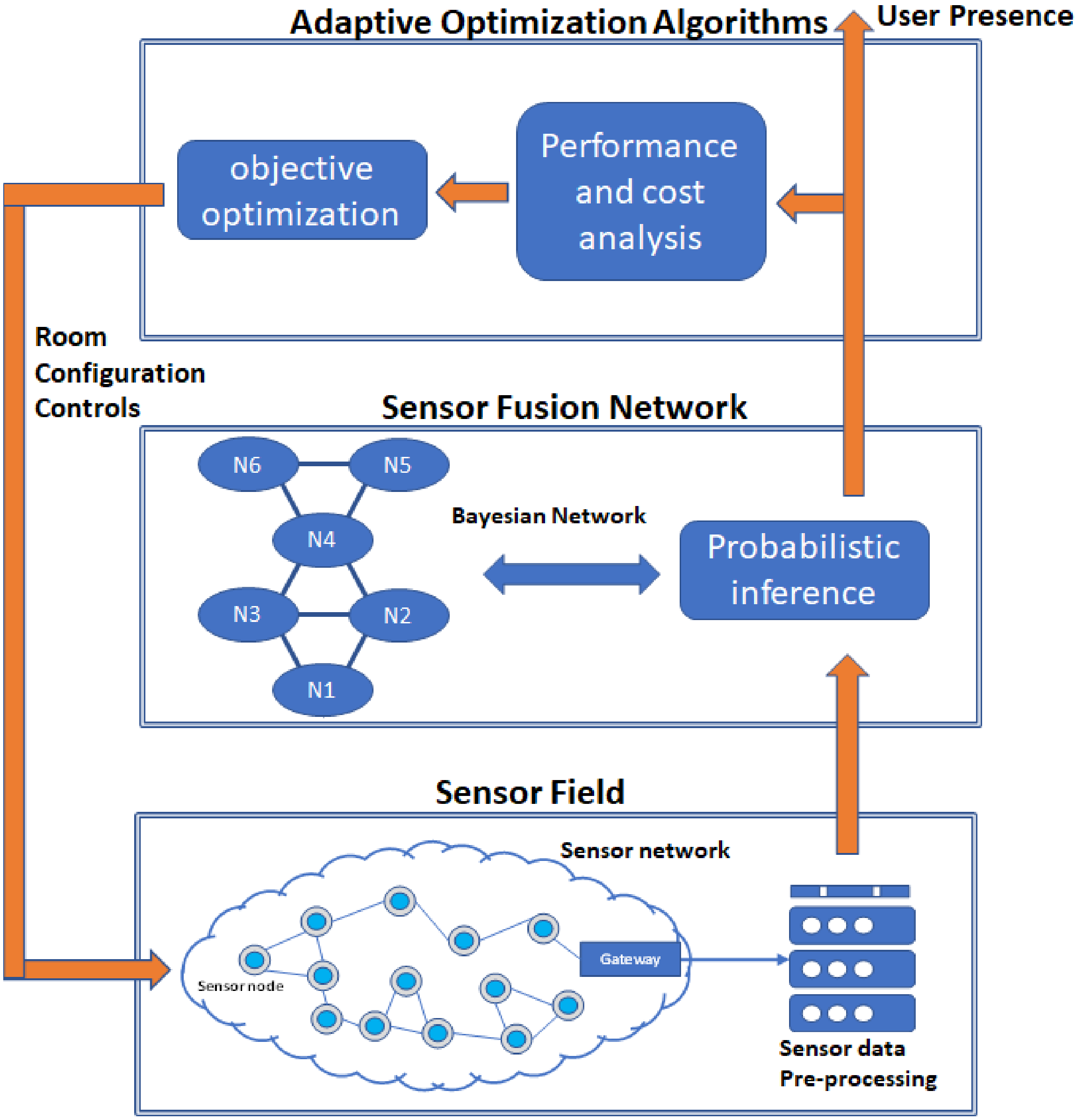


Figure 18.0 User Entry and Presence Adaptive Bayesian Sensing System [22]

Inertial Sensor Fusion [23]: An inertial navigation system is an example of a sensor fusion application where the sensors interface directly to a processor that uses the measurements in a navigation model for inertial navigation, location, and pointing. Inertial sensors for this purpose were described in Part 1, particularly the fusion of Global Positioning System (GPS) and inertial navigation system measurements within an extended Kalman filter. An inertial measurement unit (IMU) provides 3-axes of accelerometer measurements, 3-axes of gyroscope measurements, and in some cases 3-axes of magnetometer data. Magnetometer data can be used for initialization; but measurement errors can occur if it is near any platform structure with ferrite material even after calibration. Inertial navigation and location data can be obtained directly from INS/GPS stand-alone devices; however high-grade navigation devices are expensive and relatively large. An INS is really a IMU integrated with a navigation model and GPS. The INS use the sensor measurements from the IMU accelerometers and angular rate sensors in a navigation model usually implemented within the mathematical structure of an Extended Kalman Filter (EKF) with many states. The navigation model is effectively the state propagation model for the filter while also serving to define the sensor fusion architecture. There are smaller INS devices now available that can provide tactical grade heading and location information; devices such as the Gladiator Technologies LM-60 INS/GPS and LM-005 INS/GPS or VectorNav VN-100, 200, and 300. The least expensive approach is to purchase an IMU and implement the navigation model and EKF in system software. The inertial navigation model with inertial measurement unit inputs is shown in Figure19.0. Given just the accelerometer and gyro inputs the position, velocity and attitude of the body can be estimated. The accelerometers provide the linear motion measurements and the gyro angular rate data in what is termed the strapdown equation to determine and propagate the attitude matrices; these matrices are the direction cosine matrix (DCM) that define the orientation of one coordinate frame relative to another coordinate frames. Five DCMs are defined in Figure 19.0, body to inertial [CBI], body to ECEF [CBE], ECEF to inertial [CEI], NED to ECEF [CNE], and body to NED [CBN]. Attitude propagation can also be performed using quaternions. These estimates are then corrected and improved using the EKF.

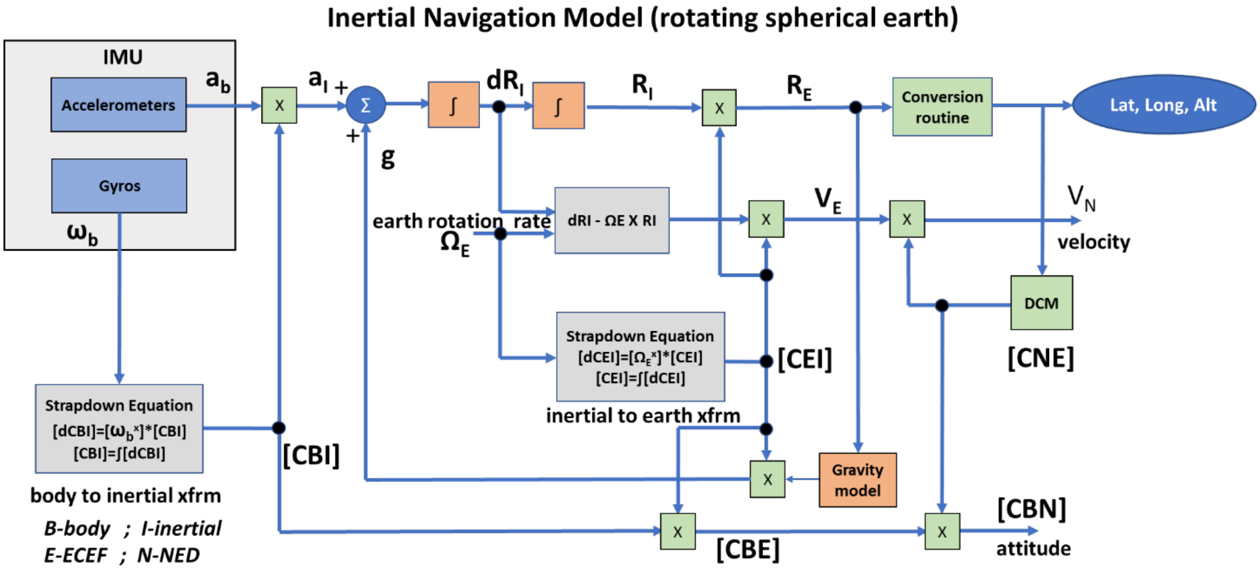


Figure 19.0 Inertial Navigation Model [23]

The output of the inertial navigation model provides position, velocity, and attitude vector inputs for the Kalman Filter as shown in Figure 20.0. Position and velocity estimates are compared with GPS measurements and the calculated error residual used for model corrections. The navigation model and IMU processing algorithms normally compensate for repeatable errors including scaling calibration, alignment, and sensor offsets.

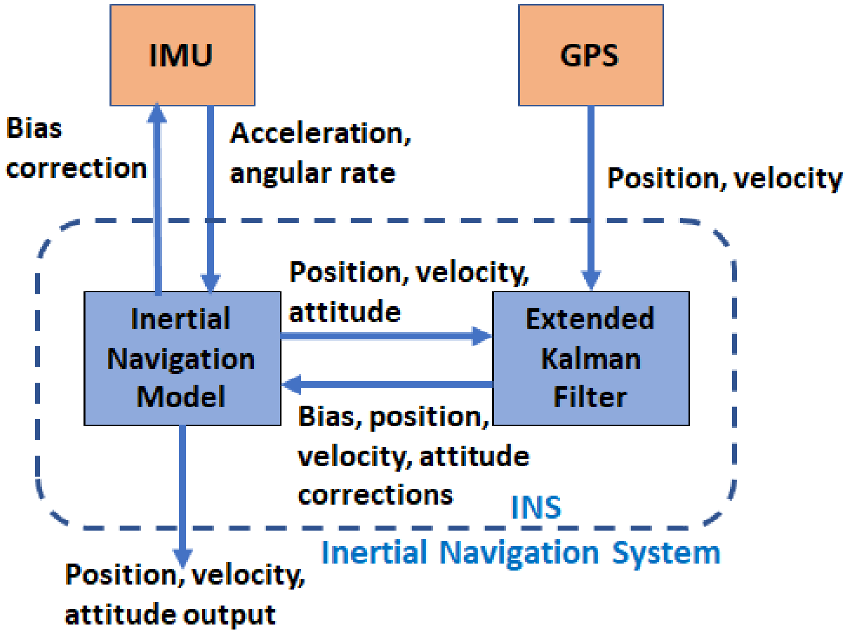


Figure 20.0 Inertial Navigation Sensing System

Regardless of the approach, whether it is a standalone IMU and GPS or a fully integrated stand-alone inertial navigation system with GPS, sensor information is fused within the framework of the navigation model and a high multi-state EKF for filtering. The primary applications for inertial sensors are navigation, inertial geo-pointing and geo-location, and stabilization of the sightline of a vehicle mounted point and track system.

Autonomous Vehicle [24]: Present autonomous vehicle technology utilizes RADARs, LIDARs, and cameras to evaluate the condition of the environment surrounding a vehicle. Sensor fusion combines data from each sensor type, using software algorithms to provide the highest fidelity and accurate environmental model possible. Each sensor type has strengths and weaknesses. Sensor fusion determines the sensor measurement information with the highest validity or belief factor to obtain the state of the environment surrounding a vehicle. Fused information can be used to generate a situational awareness map about the vehicle; identifying required actions to be taken. Measurements from all three sensor types may be fused using the redundant information from the sensors to obtain a more accurate and reliable situational awareness map.

Radars accurately determine distance and speed but don't provide a visual scene representation. Cameras provide the visual representation that provides visual information to identify objects, including pedestrians, bicycles or other vehicles. Performance is degraded, however, by dirt, sun, rain, snow or darkness. Performance is somewhat dependent on the spectral band of the camera; shortwave infrared has slightly better performance under poor environmental conditions but will still be degraded. LIDARs can accurately detect objects as well as their range. Knowing the LIDAR beam pointing angles and GPS, they can determine the object location relative to the vehicle. LIDAR in general can work over distances of miles, however for autonomous vehicles they are power and therefore range limited to between 30 meters to 50 meters due to SWaP and eye-safety constraints. They are also more expensive than cameras or RADAR. Table 1.0 provides an overview of the three primary sensor types and their strengths and weakness and the sensing system configuration is illustrated in Figure 21.0. The attitude and heading reference system (AHRS) and GPS provide inertial orientation and location information.

Vehicle-to-everything (V2X) is a vehicle to vehicle, person to person, network to network short range cellular or wireless local area network (WLAN). Connections can be combinations of these or with any communication device capable of communicating over the network, providing users a dedicated short-range communication (DSRC) link. Status and awareness information can be transmitted and received in real time which can be used to provide increased safety and reduced traffic congestion. By identifying object proximity, potential accidents can be avoided and congestion minimized by rerouting traffic patterns. V2X can also be linked with other networks that use data fusion to determine the traffic state (low volume traffic, traffic jam, high volume traffic) based on data collected from road side sensors such as acoustic, image and LIDAR data.

Table 1.0 Sensor Strengths and Weaknesses [24]

	RADAR	LIDAR	CAMERA	FUSION
Object Detection	Good	Good	Fair	Good
Pedestrian Detection	Poor	Fair	Good	Good
Weather Conditions	Good	Fair	Poor	Good
Lighting Conditions	Good	Good	Poor	Good
Dirt	Good	Fair	Poor	Good
Velocity	Good	Fair	Fair	Good
Distance Accuracy	Good	Good	Fair	Good
Distance Range	Good	Fair	Fair	Good
Data Density	Poor	Fair	Good	Good
Classification	Poor	Fair	Good	Good
Packaging	Good	Poor	Poor	Good

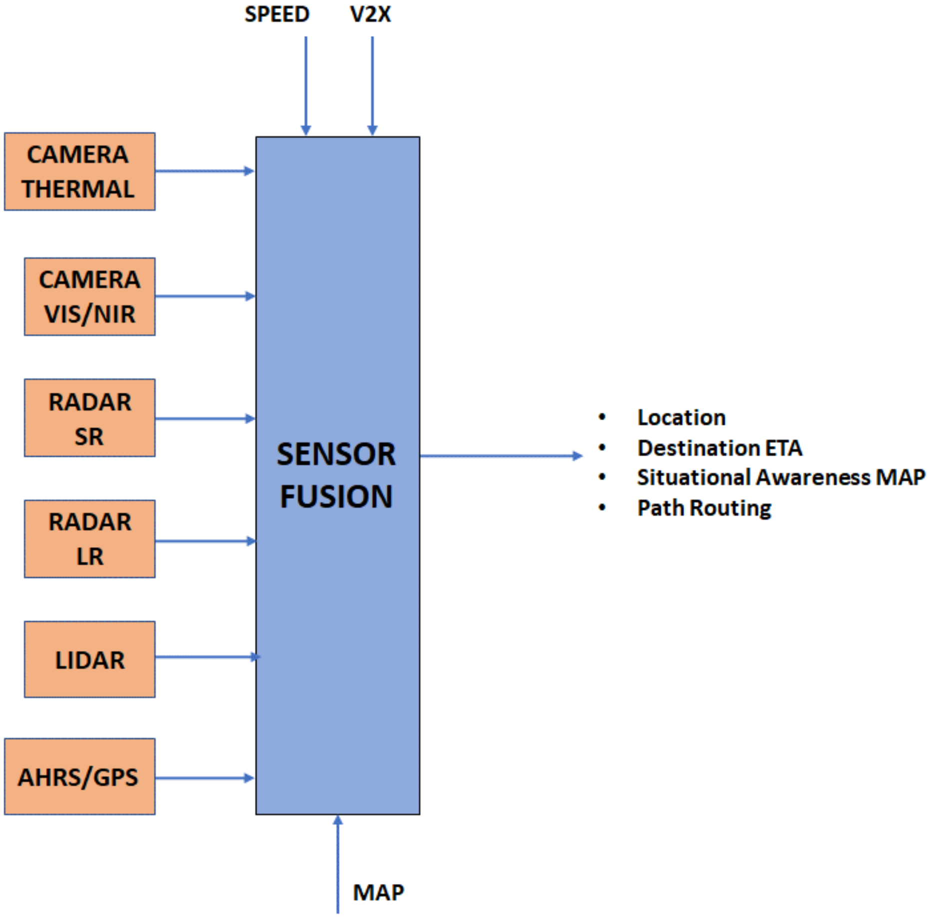


Figure 21.0 Autonomous Vehicle Sensing System Configuration

Using several vehicle sensors of the same kind with overlapping fields of regard (FOR) can provide a larger region of situational awareness coverage with greater fidelity. More than one sensor can observe or detect and confirm an object; increasing detection probability and reliability of object identification around the vehicle. This provides for a more accurate representation of the surrounding environment. With cameras, scenes can be spliced together to obtain a panoramic view of the surroundings providing improved situational awareness. Sensor fusion algorithms are actually used to support several vehicle controls functions. However multiple sensors will also drive up the cost of a sensor suite as well as increasing the burden on sensor fusion processing.

As the number of sensors increases, sensor fusion complexity increases. Algorithm designs that minimize processing delays and network communication latencies becomes more challenging. Accurate time tagging of data is required to insure the temporal alignment of data. But more vehicle sensors provide more information to improve performance so this is nearly inevitable and much of the solution may reside with architecture; more specifically centralized versus distributed sensor fusion processing, a topic already discussed. Sensor data processing and analysis for tracking objects has is often integrated with the camera or RADAR resulting in a more distributed architecture. However the centralized architecture approach provides more processing power in one high end processor located at a central node. This processor can run more complex and computationally intensive sensor fusion algorithms. With today's technology, quad-core processors and task management algorithms optimally allocate processing tasks allowing for collection of low-level sensor data that is transmitted and fused at the central processing node. This results in smaller sensors, reducing SWaP, which for a vehicle with limited space allows more options for vehicle integration and increased sensor data access. Latency is reduced since low-level measurements are transmitted immediately then fully processed at the central node as opposed to dealing with sensor processing latencies that impact data transmission speed and predictability. More data leads to better decisions. This autonomous vehicle network architecture allows for many sensors, with scalability addressed by one sensor fusion algorithm in the central processing node.

5.0 Summary Part 2

Part 2 of the course examined two sensor types very applicable to networking; MEMS and Fiber Optic sensors. This was followed by a discussion of the sensor network; specifically, the sensor networks topology and protocols. Finally, sensor fusion was described in terms of types of algorithms used and some examples. A brief review of key points from each section follow.

Section 1 Overview

- Interconnecting many low-cost MEMS sensors to a sensor network that interfaces with IoT connectivity provides the basis for collecting large amounts of data processed by sensor fusion algorithms expanding the source of knowledge much further than any single sensing system can provide.

Section 2 MEMS and Fiber Optic Sensors

- The key micro electro-mechanical system (MEMS) sensors are pressure sensors, accelerometers and gyroscopes used in consumer and medical electronics
- The basis for the MEMS technology is imbedded in the development of integrated circuit technology and the implementation of mechanical motion components on microscopic scale given the advancements in semiconductor device fabrication.
- Mechanical MEMS sensing techniques include piezoresistive, piezoelectric or capacitive with the typical quantities sensed being pressure, temperature, strain, force, rate and displacement
- Many physical quantities can be sensed optically via fiber optic sensors including light intensity, position displacement, temperature, pressure, rotation, sound, strain, magnetic and electric fields, radiation, flow, liquid level, vibration

- Fiber sensors sensing techniques can be categorized into three types, intensity modulated, phase modulated and wavelength modulated

Section 3 Sensor Networks

- A sensor network connects many sensor systems, termed sensor nodes within the network, providing measurement data on quantities such as heat, pressure, and motion, as described in Part 1.0; transmitting the information to another node or central hub.
- The network infrastructure for present day sensor networks can be wired, often using an ethernet interface, or wireless. With the rapid development of MEMS, sensor networks are a vital part of the Internet of things (IoT).
- The wireless sensor network (WSN) interconnects many nodes using wireless transmission. Radio frequency (RF) transmission is often used but can be distance limited at the sensor node level due to sensor power and size constraints
- WSN are simple to install, reconfigure, and maintain and with the on-going development of sensors communication interfaces have become a key technology of the IoT
- There are several network topologies that can be used to arrange the sensor nodes. Baseline topologies include point to point, bus, star, tree, mesh, ring, circular and grid.
- Two telecommunication systems models used for protocol design are the open systems interconnection (OSI) model and the Internet protocol suite or known as the transmission control protocol/internet protocol (TCP/IP) model.
- The protocols are defined within layered communication model structure, each layer defining a functional step for communication.
- The TCP/IP model is closest to that required for the WSN telecommunication model. Breaking out the link layer into a physical layer and a link layer would provide a typical 5-layer WSN model, or from lowest (physical) to highest (application) layer level

Section 4 Sensor Fusion

- Sensor fusion describes an algorithm or set of algorithms using data collected from many sensors and sensor types to obtain the best estimate of the state of a physical quantity.
- Sensor fusion is a term that covers a number of methods and algorithms, including: central limit theorem, Kalman Filter, Bayesian networks, Dempster-Shafer algorithm, convolutional neural networks, artificial intelligence
- Bayes' rule, based on joint probability, provides the foundation for the probabilistic models and Bayesian network to predict the likelihood that any one of several hypotheses is the contributing factor in a given event.
- Bayesian networks are ideal for predicting the likelihood of several possible causes being the contributing factor of an event that has occurred; such as the probabilistic relationships between diseases and symptoms.

- The use of the Bayesian network was applied to room access and presence security system, securing and monitoring the entry, presence and exit status within a closed facility.
- It was an adaptive system design and the architecture was composed of three functional blocks, the top block being adaptive optimization processing algorithms whose inputs came from fused data from a Bayesian network whose input came from the sensor network.
- Inertial navigation systems (INS) use a Kalman filter-based navigation model for inertial sensor fusion.
- INS inertial sensors include an inertial measurement unit containing 3 gyros and 3 accelerometers and GPS.
- The autonomous vehicle fuses camera, LIDAR, and RADAR data to obtain the best situational awareness map it can about a vehicle.
- Vehicle-to-everything (V2X) is a dedicated short range cellular or wireless local area network (WLAN) link; vehicle to vehicle, person, or network providing information for increased safety and reduced traffic congestion by identifying person and vehicle presence; avoiding potential accidents, and congestion.

References

(1) Kaivan Karimi, “The Role of Sensor Fusion in the Internet of Things” Executive Director - Global Strategy and Business Development, MCUs, Freescale Semiconductor
(2) Prime Faraday Technology Watch, “An Introduction to MEMS (Micro-electromechanical Systems)” ISBN 1-84402-020-7, January 2002
(3) https://www.sensorland.com/HowPage023.html
(4) https://en.wikipedia.org/wiki/Vibrating_structure_gyroscope
(5) Krohn D.A., Fiber Optic Sensors Fundamentals and Applications, ISA – The Instrumentation, Systems, and Automation Society 2000 ISBN 1-55617-714-3
(6) https://en.wikipedia.org/wiki/Wireless_Sensor_Network
(7) Sharma, Divya, Verma, Sandeep, Sharma, Kanika, “Network Topologies in Wireless Sensor Networks: A Review” International Journal of Electronics and Communication Technology, IJECT Vol. 4, Issue Spl - 3, April - June 2013, ISSN: 2230-7109 (Online) | ISSN: 2230-9543 (Print)
(8) https://en.wikipedia.org/wiki/OSI_model
(9) https://en.wikipedia.org/wiki/Internet_protocol_suite
(10) https://www.configrouter.com/understanding-host-host-communications-model-14206/
(11) https://en.wikipedia.org/wiki/Micro-Controller_Operating_Systems
(12) https://en.wikipedia.org/wiki/TinyOS

(13) https://en.wikipedia.org/wiki/LiteOS
(14) https://en.wikipedia.org/wiki/Sensor_fusion
(15) https://en.wikipedia.org/wiki/Central_limit_theorem
(16) https://en.wikipedia.org/wiki/Bayesian_network
(17) https://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory
(18) https://en.wikipedia.org/wiki/Convolutional_neural_network
(19) https://en.wikipedia.org/wiki/Artificial_intelligence
(20) https://en.wikipedia.org/wiki/Bayesian_inference
(21) Lockhart, Richard, "Stat 830 – Fall 2011", Simon Fraser University
(22) De Paola, S. Gaglio, G. Lo Re, M. Ortolani, "Multi-sensor Fusion through Adaptive Bayesian Networks", Networking and Distributed Systems Research Group, In AI*IA 2011: Artificial Intelligence Around Man and Beyond, 2011, pp. 360-371
(23) Stoval, Sherryl H. "Basic Inertial Navigation" Navigation and Data Link section, Systems Integration Branch, NAWCWPNS TM 8128, Naval Air Warfare Weapons Division, China Lake, Ca. Sept 1997
(24) https://www.aptiv.com/insights/article/what-is-sensor-fusion