Transformation
T-STEM 2013
New Directions | New Pathways | New Outcomes

dteach

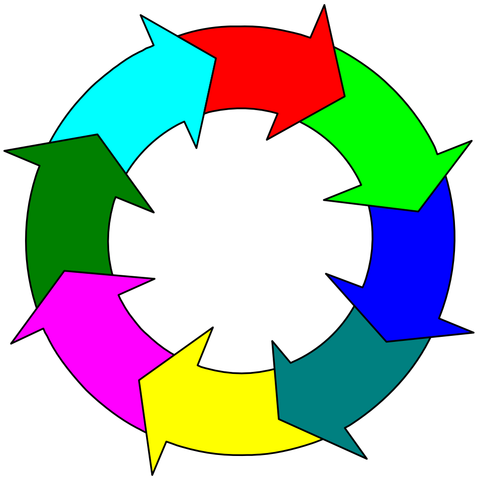
Automation and Control

Automatic machines and systems are designed to accomplish many tasks with little or no human supervision. Many examples of automatic systems are encountered everyday. Common household examples include toasters, coffee makers, microwave ovens, and air conditioning systems, some of which are shown in the figure below. Even the humble toilet is automatic, once it is flushed. One characteristic that distinguishes all of these automatic systems from other types of machines or systems is the presence of a control subsystem. In this unit, we outline the engineering concepts and technology associated with designing and manufacturing automatic systems.

Figure 1. Typical controlled systems
(www.amana.com/refrigerators/tr21V2.html ,
www.amana.com/cooking/arg7600.html,
www.ford.com, and http://www.dell.com/)

1. Control

Every automatic system has a control subsystem that determines its behavior. The control subsystem consists of two parts: a logical, or functional, description of the system and a physical manifestation of that system. The logical description, or control algorithm, describes desired system behavior, oftentimes with software. The system's physical manifestation, or hardware, consists of the actual components necessary to implement our logical description. Note that an algorithm is nothing more than a series of steps in a process that has a definite beginning and ending.



In simpler systems, the control algorithm is inherent in the components and their configuration. In effect, the systems are designed such that they behave properly. For example, with the float valve in a toilet, as shown in the figure below, water will always rise to a certain level due to the system geometry. You cannot change the float valve assembly's control algorithm, outside of minor adjustments.

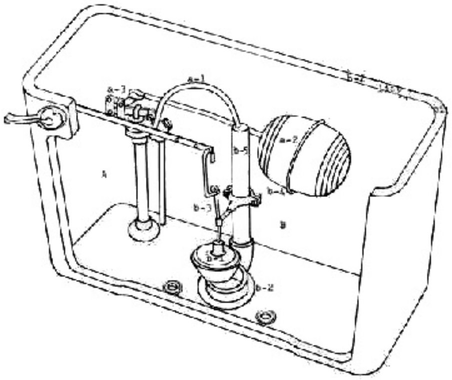


Figure 2. Typical toilet float valve assembly (http://www.toiletology.com/anatomy.shtml).

In more complex systems, a microprocessor or computer executes the control algorithm and thus controls the system. For example, modern VCRs are microprocessor-controlled which allows great flexibility in programming. There is a software program that embodies the control algorithm, and it is possible to reprogram (although not by the user).

Similarly, modern household thermostats, such as that shown below, can come equipped with microprocessors. With the capability of being programmed, homeowners can experience improved comfort and lower energy bills by increased control over their environment. The digital thermostat monitors ambient temperature, using its control algorithm to run an air conditioner or furnace in order to control the ambient temperature. Note that thermostats do not have to be electronic. For example, immersion-type thermostats, such as those found in automobile radiators, can utilize melting wax to open a valve and allow coolant flow.

Figure 3. Microprocessor-equipped household thermostat and immersion thermostats, such as from automobile's radiator
(from: www.gibsonheat.com/thermsta.htm and www3.thomasregister.com/ss/.1151955367/olc/caltherm/diesther.htm)

In contrast, you cannot reprogram a float valve. Components of the float valve system can be slightly altered, such as bending the float rod to remedy a running toilet. However, the overall control function cannot be altered.

At the extreme end, there are devices called "programmable logic controllers" that are used to run systems such as machine tools. These are meant to be reprogrammed by users and can provide great levels of capability. Two advanced applications of programmable logic controllers include the Mars Sojourner and the International Space Station (ISS), shown below. For example, the Mars Sojourner uses controllers, in conjunction with sensors, to determine its location on the Martian surface and plot a course to travel. Another example is the ISS' altitude and attitude controllers. Computers are used in conjunction with sensors and actuators to ensure the ISS' solar arrays point at the sun, and the rest of the space craft stays in position as it orbits the Earth.

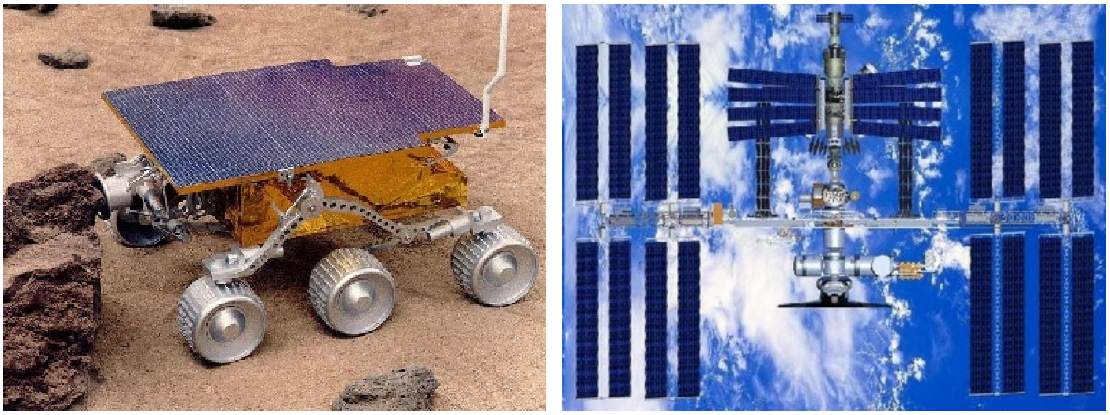


Figure 4. The Mars Sojourner and the International Space Station, two examples of highly controlled systems (www.nasa.gov).

While not necessarily capable of the flexibility inherent in commercial programmable logic controllers, the Lego Mindstorms product is another example of this increased level of programmability. With the NXT software, the Legos can be programmed with a great variety of user-created control algorithms.

(From http://mindstorms.lego.com)

2. Control Subsystem Components

In any automated system we are trying to monitor or control some **state** of the system. A state is a physical attribute of a system, such as temperature, mass, position, or velocity. For an air conditioning system, the controlled state is the room's temperature. With a toilet's float valve, the controlled state is the water level in the toilet tank. A cruise control system in a car controls the car's state, speed. A security system monitors the positions of doors and windows in a building, activating when one of those positions changes. Some systems, such as automatic washers and dryers, monitor the state of elapsed time in order to start and stop the various cycles.

Open-loop vs. Closed-loop Control In the most general sense, a control subsystem provides a certain output for a given input, as shown in the figure below. The input, or desired response, is used by the control subsystem in order to modify the output, or actual response. For example, a room temperature of 23 °C is the desired response input to the thermostat by the homeowner, and the actual response, or output, of the control subsystem is a voltage signal to the heater or air conditioner, which causes the machine to turn on. In this case, the control subsystem is being used to translate an input into a more machine-usable form.

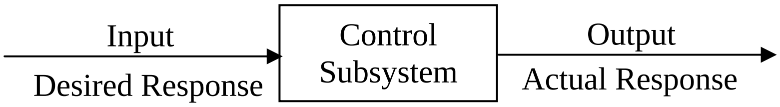


Figure 5. Simplified control system.

With the simple system shown above, an input is specified to the control subsystem, which in turn generates an output. What does this system do if the actual response is not equal to the desired response? For example, what would this control system do if our homeowner desired 23^{o}C, but got 28^{o}C? Without human supervision, in the form of changing the desired response, this system cannot compensate for the difference in temperature. This mode of operation is known as **open-loop control**, because the actual temperature cannot be used to improve the accuracy of our control subsystem. How might we improve this control scheme without human interaction?

One significant improvement is to use the actual response to modify the input to the control subsystem, this process is known as **closed-loop control**. Illustrated below, closed-loop control *feeds* the actual output *back* so that it can be used to alter the original input. This **feedback** signal is typically a measurement from a sensor.

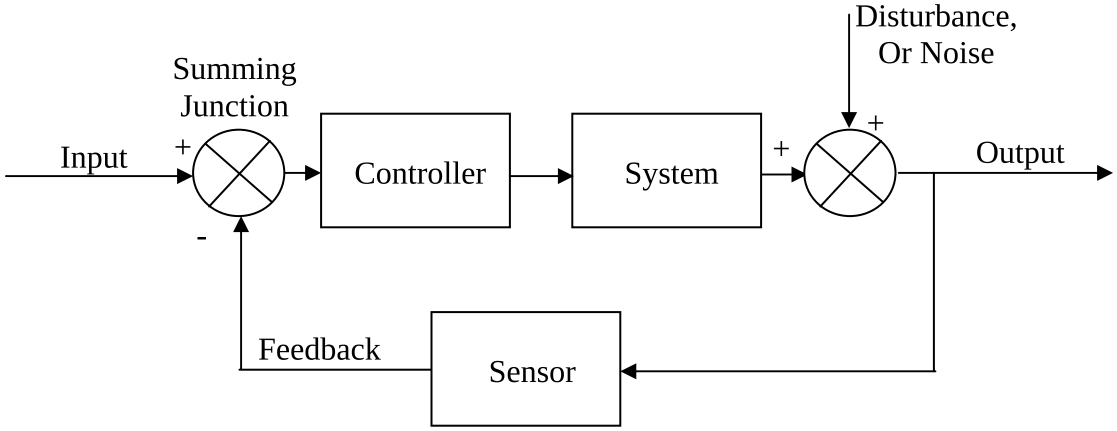


Figure 6. Example of control system with feedback and noise.

If we consider our original system of a homeowner desiring a warmer home, we again see that the **input** is our desired temperature. The **controller**, in this instance a thermostat, generates an output signal that is sent to our **system**, which is the furnace. However, we now see the presence of **noise**, or a disturbance not controlled by our system. For the heating system of a house, noise may be the children leaving the front door open. This uncontrolled heat loss will result in a lower-than-desired output temperature, unless our original input is modified.

A **sensor**, such as a thermocouple, within the thermostat measures the actual output temperature. This feedback signal is then sent back to modify our input. A **summing junction** represents the joining of two signals resulting in a single, modified signal. That modified signal represents the input signal improved by feedback taking into account the output degraded by noise. As we can see, closed-loop control systems can operate automatically, without any human interaction, in order to overcome this noise. Without such a capability, we might forever be adjusting our thermostats!

<u>Closed-loop Control System Components</u> All closed-loop control subsystems perform three basic functions: measurement, activation, and feedback or decision-making. In order to control a *state* of the system, the controller must measure the current value of the *state.* A state is any measurable attribute of a system, such as voltage, speed, or temperature. In our example above, the state is the temperature of the room. The components used to measure states are called **sensors**, and components used for activation are called **actuators**.

There are a great variety of sensors, including thermocouples and thermistors for measuring temperature, circuit breakers that measure electric current, and pressure sensors and light sensors that generate the needed feedback signals for automatic doors. Likewise, there are a great many devices that can be used as actuators: electric motors (used everywhere), valves (such as the float valve), hydraulic cylinders (used to actuate heavy machinery such as backhoes and bulldozers), and pneumatic cylinders (used to open and close dampers in commercial air conditioning systems).

The third function, **feedback**, is crucial to the implementation of any control algorithm, and is the essence of automation. Without feedback in the system, our unlucky homeowner would have to become the feedback loop and constantly adjust the thermostat in response to disturbances, or noise, such as people leaving and entering the house. Once a state has been measured, its value is fed back to the control subsystem, where the control subsystem decides what actions to perform.

<u>Closed-loop Control Applications</u> There is an underlying aspect of feedback and decision-making in even the simplest of control subsystems. In the toilet's float valve assembly, the float senses the water level in the tank and feeds that information back to the control subsystem through linkages. The assembly then uses this information to determine whether to open the valve and, if needed, by how much to open the valve.

This may seem like a stretch, as we know a float valve is not really intelligent enough to make a decision in the human sense. The physics of buoyancy and the geometric connections between the float and the valve dictate that the system will inherently act in the desired manner. Still, the notion of feedback and decision-making is a useful abstraction for designing control subsystems.

In other systems, the decision-making function is readily apparent. In the cruise control system in a car, a microprocessor performs the decision-making function. As the microprocessor receives feedback from a speed sensor, or tachometer, the processor determines the correct throttle position necessary to maintain the desired speed. One potential actuator in this case is a solenoid that moves the throttle arm back and forth.

Another microprocessor-governed closed-loop system example is today's modern 35mm camera, such as that shown in the figure below. Many current cameras have auto-focusing capabilities, using an infrared sensor to determine the distance to a photographic subject. This sensor information is then fed back to the controller and used to start actuators that focus the camera's lens, as necessary.

Figure 7. An example of microprocessor-governed, closed-loop control;
the modern day 35mm camera
(http://www.nikonusa.com).

<u>Signal/Information Flow in Closed-loop Control</u> The discussion above indicates that a control subsystem is essentially an information processing system. Sensors gather information about the controlled system's state and transmit that information to the controller. The controller then sends information to the actuators, which perform the appropriate actions on the system in order to alter the original state.

Information in our control systems is typically represented by **signals**, which can be generally thought of as energy flows. As indicated above, many different types of energy can be used as signals in control subsystems. The most common way to represent and transmit signals today is with electricity. Electrical sensors are typically small, accurate, and inexpensive, with the signals typically requiring low power levels. These low power requirements, combined with electrical energy which is abundantly available through avenues such as batteries and wall outlets, result in a very significant use of electrical signals in today's world.

Consider our example above, with the homeowner's thermostat controlling their residential central air conditioner and heater system. The thermostat is essentially a temperature sensitive switch, typically consisting of a bi-metallic strip, made of two strips of different metals joined back-to-back. In a bi-metallic strip, when the temperature of the strip changes the two metals expand and contract by different amounts, causing the strip to bend. This bending action is then used to open and close a set of electrical contacts. When the contacts in our thermostat close, a circuit is completed with a heavy-duty relay coil, known as a contactor. When the contacts on the contactor close, power is sent to the air conditioner's compressor and fan motors, starting the cooling process, or to the furnace's heating element and fan motors, starting the heating process.

The object of this discussion is to point out that the sensor (the thermostat) sends a signal to the actuator (the contactor) by way of an electrical circuit. This control circuit operates at a much lower voltage, typically 24 volts, than the compressor and fans, which are typically at 230 volts. In this control subsystem a low power signal is used to control a high power energy flow, which is a typical use of control subsystems. Sensors generate low power signals, and the actuators convert the low power signals into high power actions.

The examples above also illustrate that many different types of energy can be used in a control subsystem, with electrical energy being the most common, as discussed above. An alternative to electrical energy can be found in commercial air conditioning systems, which often send signals pneumatically, with compressed air. The system's thermostat varies air pressure as the temperature varies, and the air pressure actuates a pneumatic cylinder, which pushes a piston connected to a damper in the duct.

Many position sensors are mechanical or magnetic. In an automobile engine, cams on the camshaft mechanically indicate the crankshaft's position to the valves, telling the valves when to open and close. In the not-too-distant past, a cam lobe on the distributor shaft, connected to the crankshaft through gears, was used to tell the spark plugs when to fire. In most modern vehicles, this has been replaced by a magnetic pickup that senses magnetized areas on the distributor shaft.

3. Control Logic

The design of an automatic system is challenging. The engineer must design not only the main system that does work or performs the primary task, but also the control subsystem. The control subsystem's design begins with the control algorithm. However, the control algorithm is not really a physical entity – it is actually a process, or functional description, as discussed above. How do we represent a control algorithm in order to facilitate its design?

<u>Types of Representations</u> There are many ways to represent control algorithms. In devices controlled by timers, a timing diagram forms the basis of any control algorithm.

For example, this is a common method to represent control flow for automatic washers. Flow charts are another method to represent control algorithms and are commonly used in the of design computer programs. Similarly, flow charts can be used to design control algorithms to be executed by microprocessors.

<u>Flow charts</u> In this unit, we present another method for representing control flow – the flow chart. Flow charts are a graphical representation of control flow that explicitly models sensing, activation, and decision-making. The graphical interface of NXT Software is very similar to the flow charts illustrated below.

Flow charts use different symbols to represent the different functions in a control algorithm. The symbols we will use are illustrated in Figure 8, although other symbol choices are possible.

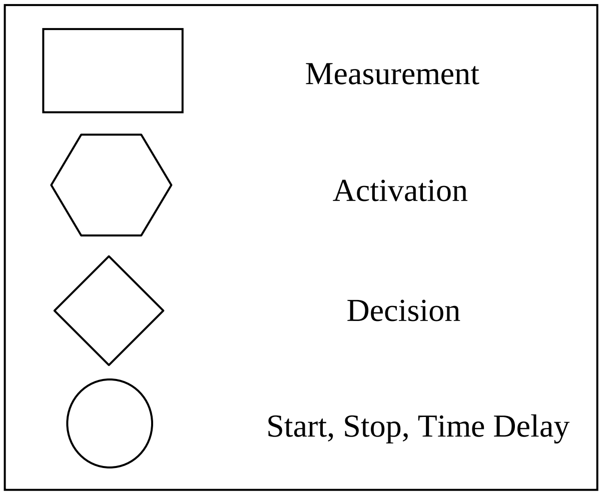


Figure 8. Symbols we will use in flow charts.

In Figure 8, “Measurement” represents a sensing step such as our thermostat measuring a room’s current temperature to be used as a feedback signal. “Activation” indicates an action that is initiated by our control system. For our cold homeowner example, activation in the thermostat control system occurs when the furnace and blowers are started. “Decision” represents a choice the control system must make, such as whether heating is required or not in the case of our homeowner. Lastly, “Start” indicates the beginning phases of our algorithm, while “Time Delay” represents a pause in the system.

A flow chart is constructed by stringing needed symbols in the appropriate sequence. Each symbol is labeled to indicate its specific function for the control subsystem being designed. For example, Figure 9 shows a flow chart for a toilet float valve, with each symbol labeled with its specific function. For instance, the measurement block representing the float ball is labeled “measure water level” to indicate that it measures the water level. The measured signal, or feedback, is then used later in the algorithm to alter the tank’s state.

Symbols in a flow chart are connected with arrows to indicate the overall flow of a control algorithm, i.e., the sequence in which functions are performed. Arrows may flow forward or backward, oftentimes creating loops so that a function or sequence of functions is performed more than once. This is also typically used with decision boxes, where the state is continually measured until it reaches a certain value. This is readily seen in Figure 9, wherein the water level is measured repeatedly, with the sensor's signal feeding into the decision box until water is needed in the tank.

Every flow chart has a start and stop symbol, indicating the start and end of the control flow. In the float valve example, the algorithm starts by measuring the tank's water level. The control subsystem then "decides" whether the tank is full or not. This decision process is represented by the decision box, labeled, "tank full?"

If the tank is indeed full, the arrow annotated, "yes", is followed and the water level measurement is repeated. This decision choice is actually caused by the float ball dropping in response to a lower tank water level. If, upon re-measurement, the tank level is not full, the arrow denoted, "no", is followed and the flapper valve is opened. Water then flows into the tank while another the water level is monitored by the second "measure water level" box. In essence, the float now senses the water level in the filling tank with the subsystem again deciding whether the tank is full or not. If the tank is not full, the subsystem continues to measure the water level. Once the answer is "yes", the tank is full and control flow proceeds to the "close valve" activation box, leading ultimately to the control algorithm's end.

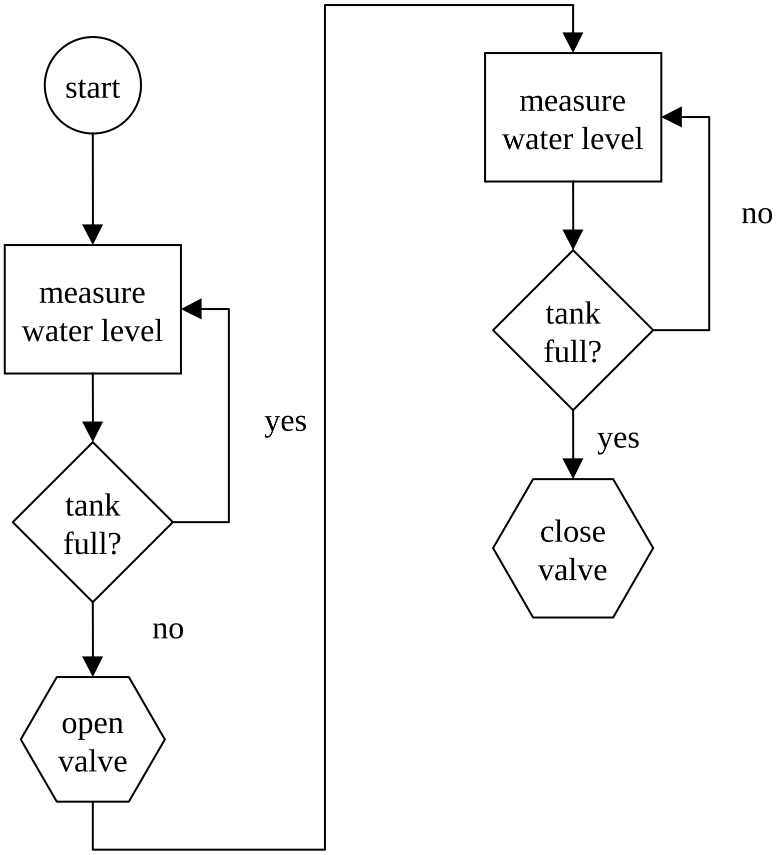


Figure 9. Flow chart for a toilet's float valve subassembly.

This can easily seem like an overly complex representation of what a float valve actually does. For instance, does a float valve really decide if the tank is full? Arguably, the valve automatically closes when the float reaches a certain level, and there is not really any intelligence in the device capable of making a decision. However, our flow chart is a useful abstraction for design purposes.

We can indeed design a new tank filling control subsystem that has a separate water level sensor to provide feedback to a microprocessor. The microprocessor can then make the decision as to whether the valve needs opening or closing. Remarkably, this new subsystem would have the same flow chart! Flow charts are *functional* descriptions of control subsystems. As such, they are independent of form and allow an engineer to make choices about actual components, or hardware, used to implement a given control subsystem.

Figure 10 shows a control subsystem flow chart for a gas-fired central furnace used to heat a home. As with our homeowner example above, the state sensed by the controller is temperature in the home, with a thermocouple/thermostat used as the sensor. The control algorithm begins with the thermostat sensing the temperature. Control then flows into a decision box, which determines whether the temperature has dropped below the **set point** of the thermostat. The set point is the state value that the control subsystem is trying to maintain. Just as our homeowner originally desired a temperature of 23°C, the desired room temperature is again our target.

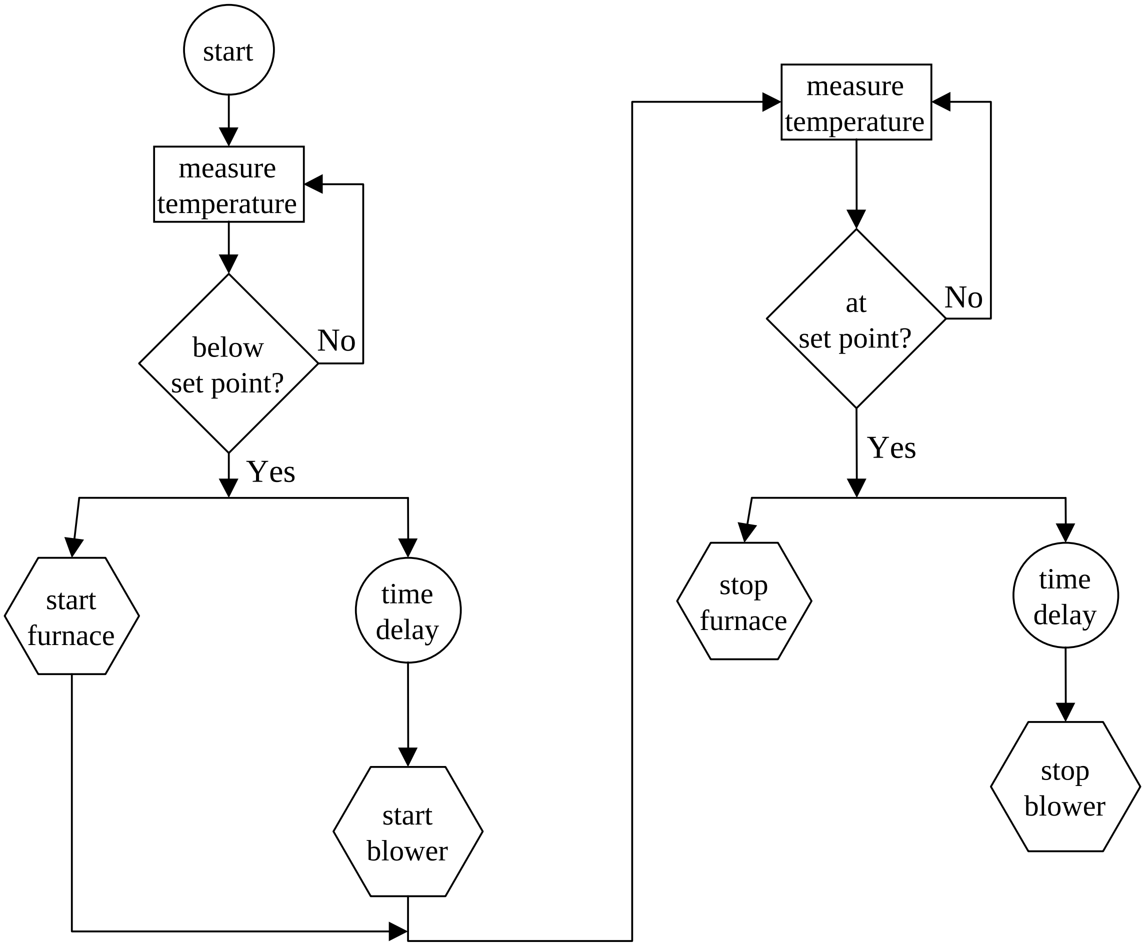


Figure 10. Flow chart of a natural gas-fired furnace.

Once the controller determines that the temperature is too low, the control algorithm gets interesting. Notice that the control flow splits off into two sub-flows. The first one activates the furnace, in a straightforward manner. However, in a gas-fired furnace, the heat exchanger (the component that transfers heat from the combustion chamber to the air) is typically cold when the furnace first comes on. If the blower were to start immediately, cold air would be blown into the conditioned space of our home. To avoid this, most gas-fired furnaces delay blower start-up to give the air in the heat exchanger time to warm up. This is reflected in the flow chart by splitting the control flow.

After the furnace and blower are both activated, the control flow merges. The controller then measures the room temperature until it reaches the set point. At this point, the control flow splits again. After the furnace shuts off, there is still warm air in the heat exchanger. For safety purposes, the blower usually continues to operate to cool the heat exchanger. This is indicated on the flow chart by the time delay symbol before the blower is stopped. After the blower stops, the control flow again merges, and the control algorithm ends.

4. Design of Automated Systems

Design of automated systems is complicated by the need to design the control subsystem. The requirements of the control subsystem are intricately tied to the way the overall system must function. This situation calls for a holistic approach to system design, in which the overall function and subfunctions of the system as a whole are clearly identified. This is the need-function-form approach to design that we advocate for DTEACh. In this section, we will point out how this approach applies to the design of the control subsystem.

The first step in designing the control subsystem is the design of a control algorithm, which is essentially a step-by-step description of the functions that the control subsystem must perform. A flow chart is one of the many ways to represent the control algorithm. Follow the steps outlined below to design a control algorithm using a flow chart.

1. Identify the states of the system to be measured and controlled.
2. Identify the actions to be performed in response to state changes.
3. Lay out the measurements and actions in sequence.
4. Add decision boxes at appropriate places in the sequence. Every measurement is usually followed by a decision box, with the decision being what to do in response to the measurement that has been fed back.
5. Add in time delay symbols where necessary.
6. Connect the elements of the flow chart with arrows to indicate the control flow. Where actions or measurements must occur at the same time (parallel), split the control flow.

As with any design problem, design of the control algorithm is an iterative process. You will probably perform each of these steps more than once. As you experiment with the control subsystem, you may need to redesign the control algorithm by adding in other actions or measurements, or rearranging the sequence or control flow. Don’t be discouraged, as this is part of the design process.

The next phase of designing the control subsystem is choosing the components. This means you must choose sensors and actuators and decide how signals will be transmitted between the two. What type of energy will you use? How will decisions be made? Will you have a separate programmable microprocessor such as the NXT Brick used with NXT Software, or will you construct a system such as the toilet float valve that is designed to inherently follow the control algorithm? These decisions are similar to the ones engineers must make when designing any technical system, and must be made with knowledge of available technology, as well as with consideration for power requirements, cost, and manufacturability. The engineer must also consider how the control subsystem integrates with the overall system when making component choices.

Do you currently implement control theory in your classroom?