interfaces, and one or more processors. Before delving into this complexity, let's review the elements of a service as shown in Figure 1.2.
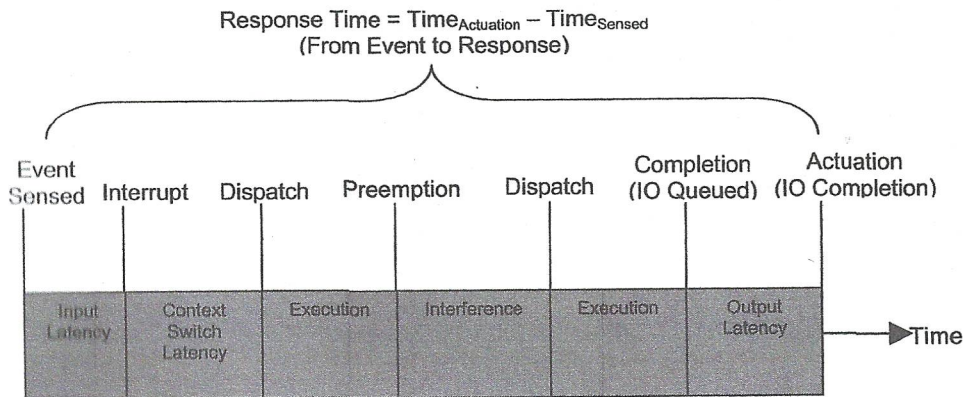
$$\text{Response Time} = \text{Time}_{\text{Actuation}} - \text{Time}_{\text{Sensed}}$$
(From Event to Response)

FIGURE 1.2 Real-Time Service Timeline

Figure 1.2 shows a typical service implemented with hardware IO components, including analog-to-digital converter interfaces to sensors (transducers) and digital-to-analog converter interfaces to actuators. The service processing is often implemented with a software component running as a thread of execution on a microprocessor. The service thread of execution may be preempted while executing by the arrival of interrupts from events and other services. You can also implement the service processing without software. The service may be implemented as a hardware state machine with dedicated hardware processing operating in parallel with other service processing. Implementing service processing in a software component has the advantage that the service may be updated and modified more easily. Often, after the processing or protocol related to a service is well known and stable, the processing can be accelerated with hardware state machines that either replace the software component completely in the extreme case or, most often, accelerate specific portions of the processing.

For example, a computer vision system that tracks an object in real time may filter an image, segment it, find the centroid of a target image, and command an actuator to tilt and pan the camera to keep the target object in its field of view. The entire image processing may be completed 30 times per second from an input camera. The filtering step of processing can be as simple as applying a threshold to every pixel in a 640 × 480 image. However, applying the threshold operation with software can be