Dependable Cyber-Physical Systems: a Software-based Approach

Junsung Kim
Carnegie Mellon University

CPS enable various new applications, including drones, implantable medical devices, smart cars, distributed transportation systems, smart grids, and planetary robots. A recent report from National Institute of Standards and Technology highlighted that the technical CPS innovations could be applicable to areas constituting up to $82 trillion in economic activity by 2025. As CPS become part of everyday life, we will have many societal benefits ranging from autonomous driving preventing accidents to smart buildings saving energy to implantable medical devices changing the paradigm for patient treatment.

The rise of CPS, however, poses new dependability challenges (among others). CPS sense the physical environment, process data in real-time, control the actuators, and guarantee the timing of the whole execution chain for ensuring safety. Since CPS are tightly coupled with the physical world, anomalies such as hardware failures and timing errors may cause significant damage to life and/or property. Common practices addressing those failures tend to over-provision resources, replicating hardware components and keeping CPU and network loads low. Many CPS systems are targeted towards large-scale cost-sensitive markets that have stringent space and bill-of-material constraints that cannot afford overprovisioning. For example, the automotive industry has been trying to consolidate in-vehicle CPUs to reduce assembly and maintenance costs, as CPU and network hungry autonomous driving features hit the market. To tackle such challenges, I propose adaptive graceful degradation and smart use of sensor/actuator modalities.

A. Adaptive Graceful Degradation

Graceful degradation is a well-established approach to maintain limited functionality in a system with a component failure. The basic idea behind is to avoid a potential undesirable event by providing restricted features to accommodate the reduced resources due to a failure. When it comes to autonomous vehicles, for example, graceful degradation should be appropriately adjusted depending on different situations. Suppose a failure on a processing board running vision algorithms to detect pedestrians. If a vehicle with the failure is driving on a highway, the vehicle may notify its driver of the failure and keep driving. If the vehicle is in an urban area, pedestrians are highly likely to be present. Hence, the vehicle may run the pedestrian detection algorithms in a degraded mode (possibly with a low frame rate) on another live processing board and also slow down the vehicle. It is important to apply graceful degradation in an *adaptive* manner so that a system can recover a failure even with less resources.

In CPS such as autonomous vehicles, most algorithms (tasks) deal with a periodic sequence of perception, computation and control. The periods of such tasks play an important role in determining how much resources are required in the system. By adjusting task periods, we can regulate *utilization*[1] that can be treated as workload. Lowering the utilization of a task creates more room for other tasks to use. In other words, a framework for *adaptive graceful degradation* is indispensable to run critical tasks with limited resources caused by a failure. For example, the vision algorithms mentioned above can be run on another live processing board along with tasks that are adjusted to have lower utilization.

B. Smart Sensor/Actuator Control

Considering sensor and actuator failures is vital for dependable CPS. Since many accidents in avionics and automotive industries can be traced to unexpected sensor failures, how CPS can be tolerant to them should be investigated. CPS with cost and space constraints may not always be able to have redundant sensors. Thus, different sensor modalities can be leveraged when sensors, such as cameras and radars, have overlaps in their vantage points. I will identify failure conditions for each sensor or actuator type and develop methods to detect failures. Any detected failure will be notified to higher layers so that algorithms can be reconfigured to use fewer sensors or actuators while generating less accurate but useful outputs.

[1]The utilization of a task is defined as the ratio of worst-case execution time of a task to its period.

Many analog sensors are also prone to intermittent faults, so using different sensor modalities is better than duplicating the same type of sensors because different types of sensors typically react to the same environmental condition in diverse ways. Suppose a vehicle is equipped with radars for blind spot detection. If a backward-looking radar does not work properly, a vision algorithm detecting obstacles from images obtained through a backward-looking camera can be used. A similar approach is applicable to actuators, too. An autonomous vehicle may use a low-grade sensor with complex data-processing algorithms after a high-grade sensor with simple algorithms fails, until the vehicle can safely stop.

C. System-level Architecture for Failure Evasion in Real-time applications

Realizing adaptive graceful degradation and smart use of sensor/actuator modalities requires a runtime framework with *flexible* configuration options. I designed and implemented a layer called System-level Architecture for Failure Evasion in Real-time applications (SAFER) [1]. It incorporates configurable software mechanisms and policies to tolerate failures of critical CPS resources while meeting task timing constraints. SAFER supports the fault-tolerance schemes of hot standby, cold standby, and re-execution. It also fuses and (re)configures sensor data used by tasks to recover from system failures. The proposed system was evaluated on an autonomous vehicle automated by our team at Carnegie Mellon [2] and showed that processor failures did not affect the autonomous driving quality when SAFER was enabled.

SAFER can be easily extended to support adaptive graceful degradation. When a processing board failure is detected, the standbys of the primary tasks on the failed board can be activated to run elsewhere. If resources are limited, SAFER will make sure that all required tasks are executed in a degraded manner. The schedulability of the adjusted tasks can be guaranteed by using admission control algorithms or response-time tests that can handle varying periods [3]. Depending on a given condition, the best configuration parameters can be adaptively set by SAFER.

SAFER can effectively use sensor/actuator modalities. The SAFER layer should have the capability to detect sensor anomalies that are different for each type of sensors, which can be a plug-in module for the SAFER layer. When a sensor failure happens, SAFER can trigger a different configuration using different types of sensors to recover from the failure. Since different data-processing algorithms are mandatory for different types of sensors, the logical combination among algorithms are given *a priori* as configuration parameters. Then SAFER layer can assign suitable amount of resources.

D. Potential Impact and Further Work

Moving forward, I will enable large-scale CPS to operate reliably. One of my future approaches is to accomplish this by addressing the lack of global perception. As CPS stand now, each node constituting large-scale CPS can only access local sensors. I will leverage communications as a sensor to make local sensory information globally available. In intelligent transportation systems, for example, it will generate an occlusion-free perception system for safety (avoiding accidents), lower delays (avoiding congested routes), and fuel efficiency (avoiding sudden acceleration/deceleration). In virtual hospitals, a remote surgeon will be able to perform a surgery. Cooperative citywide surveillance systems will be able to find missing kids, prevent theft and robbery, and rescue people in danger. To this end, timely and reliable interactions among the CPS nodes will play a major role. The effects of any resource failures on the entire system must be minimized by isolating such failures. I am also interested in how CPS co-exist with people where uncertainties arise not just from the physical environment but also from the people operating the system.

REFERENCES

[1] J. Kim, G. Bhatia, R. R. Rajkumar, and M. Jochim, "SAFER: System-level Architecture for Failure Evasion in Real-time Applications," in *IEEE Real-Time Systems Symposium*, 2012.

[2] J. Wei, J. Snider, J. Kim, J. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a Viable Autonomous Driving Research Platform," in *IEEE Intelligent Vehicles Symposium*, 2013.

[3] J. Kim, K. Lakshmanan, and R. Rajkumar, "Rhythmic Tasks: A New Task Model with Continually Varying Periods for Cyber-Physical Systems," in *IEEE/ACM Third International Conference on Cyber-Physical Systems*, 2012, pp. 55–64.