Reliable CPS Design for Unreliable Hardware Platforms

Wanli Chang, Swaminathan Narayanaswamy, Alma Pröbstl,
and Samarjit Chakraborty

1 Introduction

Battery-operated cyber-physical systems (CPS) increasingly exist in households, factories, and the public area. For instance, zero local emission, independence from fossil fuels, and potential improvement of energy conversion efficiency have made electric vehicles (EVs) an alternative of conventional vehicles with internal combustion engines (ICEs). Design of the underlying embedded control loops such as electric motor control, braking control, stabilization, and battery management plays a crucial role in EVs and other types of battery-operated devices. Conventionally, these control loops are evaluated by a number of quality-of-control (QoC) indices. One common QoC metric is settling time. In order to ensure performance and reliability, the design also needs to take into account a number of issues on the hardware implementation platforms, such as battery behaviour and semiconductor aging. Battery is the key component influencing the device performance, when being the main power source. As the integrated circuit fabrication technology has progressed, processors become more and more susceptible to aging. In order to ensure correct functioning, the processor operating frequency has to be reduced, which could potentially worsen QoC and compromise reliability. The focus of this

W. Chang (✉)
University of York, York, YO10 5DD, UK
e-mail: wanli.chang@york.ac.uk

S. Narayanaswamy · A. Pröbstl
TU Munich, München, Germany
e-mail: swaminathan.narayanaswamy@tum.de; alma.proebst@tum.de

S. Chakraborty
UNC Chapel Hill, Chapel Hill, NC, United States
e-mail: samarjit@cs.unc.edu

chapter is on a design framework towards reliability of CPS considering unreliable hardware platforms.

A battery pack with large capacity is needed to offer longer usage. However, with larger capacity, the battery weight also increases leading to higher energy consumption. Moreover, the capacity is often restricted by the space that can be allocated to the battery pack. One potential solution to the above problem is to design the controller in such a way that the energy consumption of the control task can be minimized.

All off-the-shelf battery packs are labelled with a nominal capacity. However, due to the rate capacity effect, the full charge capacity (FCC) of a battery pack, which is defined to be the amount of electric charges that can be delivered from the battery after it is fully charged, actually varies with different discharging current profiles. Generally speaking, larger discharging current tends to reduce the FCC. For most common lithium-ion batteries in the market, the capacity could potentially get significantly compromised if the rate capacity effect is not properly considered in the control systems design. In this chapter, we discuss an optimization framework considering QoC as one design objective and battery usage as the other. We quantify the battery usage by the total duration that the battery can be used to continuously run the control task after one full charge. In order to maximize the battery usage, the energy consumption of the control task should be small and the battery FCC should be increased by generating a battery-friendly discharging current profile. The battery aging effect can also be incorporated. That is, the battery behaviour in the long run is another optimization dimension.

The other important design aspect is processor aging. As a processor ages, the switching time of its transistors increases, resulting in longer path delays. On-chip monitors could be used to measure the delay of the critical path. It always has to be guaranteed that the signal transmission can be completed along any path within one clock cycle. Therefore, the processor operating frequency is reduced based on the new critical path delay. On the other hand, a shorter sampling period can potentially provide a better QoC. Therefore, with a smaller processor operating frequency, the sampling period increases and QoC gets deteriorated, which is dangerous and thus highly unwanted for safety-critical applications such as electric motor control in EVs. To deal with the above situation, we can re-optimize the controller with the longer sampling period, which results from processor aging.

The entire design flow towards CPS reliability considering unreliable hardware platforms is divided into two phases. In Phase I, before the processor ages, an optimization framework is used with QoC and battery behaviour considered as design objectives. With heuristic methods implemented, this battery-aware controller design gives a Pareto front of well-distributed and non-dominated solutions. The trade-off between these objectives is explored. In Phase II, after the processor ages, QoC is found to get degraded if the controller design does not change. The same optimization framework is used with slight modification. The processor aging effect is mitigated in the way that there is a minimal change of QoC with all safety requirements satisfied.

The remainder of this chapter is organized as follows: Sect. 2 gives an overview of the background on embedded control systems design, battery rate capacity effect

and aging, as well as processor aging. In Sect. 3, we present the reliable CPS design framework, and finally, Sect. 4 concludes the chapter.

2 Background

2.1 Control Systems

In this subsection, we first describe the feedback control application considered in this chapter and several background concepts. Then, we present the system modelling of the electric motor control application in EVs.

2.1.1 Basic Concepts

Plant Dynamics A control scheme is responsible for controlling a plant or dynamical system. In this chapter, we consider linear time-invariant (LTI) single-input single-output (SISO) systems,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \\ y(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^m$ is the state vector, $\dot{\mathbf{x}}(t)$ is the derivative of $\mathbf{x}(t)$ with respect to time, $y(t)$ is the output, and $u(t)$ is the control input applied to the system. The number of dimensions for the system is m . Constant matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are of appropriate dimensions with respect to m . In a state-feedback control algorithm, the control input $u(t)$ is computed utilizing the plant states $\mathbf{x}(t)$ as feedback signals. The computed $u(t)$ is then applied to the plant, which is expected to achieve the desired behaviour.

Discretized Dynamics In most applications, the controller is implemented in a digital fashion on a computer. This implies that the plant states must be sampled when measured by the sensors. Assuming the sampling period to be a constant, the continuous-time system in (1) can be transformed into the following discrete-time system:

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A}_d\mathbf{x}[k] + \mathbf{B}_d u[k], \\ y[k] &= \mathbf{C}_d\mathbf{x}[k],\end{aligned}\tag{2}$$

where sampling instants are $\{t_k \mid k \in \mathbb{N}\}$, the sampling period is $t_{k+1} - t_k = h$, and

$$\mathbf{A}_d = e^{\mathbf{A}h}, \quad \mathbf{B}_d = \int_0^h \left(e^{\mathbf{A}t} d\mathbf{t} \right) \mathbf{B}, \quad \mathbf{C}_d = \mathbf{C}.\tag{3}$$

It is noted that $\mathbf{x}[k]$ and $y[k]$ are the values of $\mathbf{x}(t)$ and $y(t)$ at $t = t_k$. The initial condition is denoted as $\mathbf{x}[0]$. The control input $u[k]$ is applied to the plant from t_k to t_{k+1} .

Feedback Controller One common goal of a control task is to make $y[k] \rightarrow r$ as soon as possible, where r is the reference for $y[k]$ to track. Towards this and other application-specific objectives, we design $u[k]$ utilizing the states $\mathbf{x}[k]$. This is then a state-feedback controller with a general structure as follows:

$$u[k] = \mathbf{K}\mathbf{x}[k] + Fr, \quad (4)$$

where $\mathbf{K}$ is the feedback gain vector and F is the feedforward gain.

Closed-Loop System With the feedback controller as shown in (4), the system closed-loop dynamics from (2) becomes

$$\mathbf{x}[k+1] = (\mathbf{A}_d + \mathbf{B}_d\mathbf{K})\mathbf{x}[k] + \mathbf{B}_dFr = \mathbf{A}_{cl}\mathbf{x}[k] + \mathbf{B}_dFr, \quad (5)$$

where $\mathbf{A}_{cl}$ is the closed-loop system matrix. Different locations of the closed-loop system poles, i.e., eigenvalues of $\mathbf{A}_{cl}$, result in different system behaviours. In the pole placement, poles are placed (eigenvalues are set) to fulfil various high-level goals, such as optimization of QoC and other application-specific objectives, and constraints satisfaction. In order to ensure system stabilization, all the poles must be less than unity. In this chapter, we restrict the poles in the real non-negative plane—which is common in most of the real-life design problems.

Once the poles are decided, the feedback gain vector $\mathbf{K}$ can be computed with Ackermann's formula. The static feedforward gain F used to make $y[k]$ track the reference r can be computed by

$$F = \frac{1}{(\mathbf{C}_d(\mathbf{I} - \mathbf{A}_{cl})^{-1}\mathbf{B}_d)}, \quad (6)$$

where $\mathbf{I}$ is an n -dimensional identity matrix [2].

QoC We use settling time as the metric to quantify the QoC. The time it takes for the system output $y[k]$ to reach and stay in a closed region around the reference value r (e.g., $0.98r-1.02r$) is the settling time and denoted as t_s . Shorter t_s indicates better QoC. When the controller poles are given, and the feedback and feedforward gains are computed accordingly, the output behaviour can be simulated, and the settling time can be derived.

Constraints There are often hard physical constraints that have to be respected in the embedded control systems, as part of the safety requirements [6, 8]. For instance, the input signal $u[k]$ could be constrained by an upper limit $U_{\max}$ and a lower limit $U_{\min}$. Similarly, the plant states could be constrained by a region. With the given controller poles and the corresponding gains, both the plant states and the control input throughout the entire control task can be simulated. Therefore, the constraints satisfaction can be evaluated.

Fig. 1 A diagram of a DC motor with the armature circuit powered up by a battery pack

Fig. 2 Pulse-width modulation control signals in the armature circuit to adjust the DC voltage applied to the rotor

2.1.2 Electric Motor Control

Electric motor control is a key function in EVs. As shown in Fig. 1, we consider a DC motor running in the *speed control* mode. The controller is supposed to operate the motor at various speeds according to the driver input and environmental conditions. The DC voltage provided by the battery pack is V . The resistance and inductance in the armature circuit are R and L , respectively. The back electromotive force (EMF) from the motor is e . The insulated gate bipolar transistor (IGBT) works as a switch controlled by pulse-width modulation (PWM) signals at the gate. When the switch is on, V is applied to the armature circuit. When the switch is off, the diode flows out the remaining current in the motor and thus the applied voltage is equivalent to zero. Periodic PWM signals are shown in Fig. 2, where the duty cycle c is calculated as

$$c = \frac{t_{on}}{t_{period}}, \quad (7)$$

and the effective voltage applied in the armature circuit is

$$V_{eff} = cV. \quad (8)$$

We can clearly see that V_{eff} is adjustable between 0 and V by controlling the PWM signals.

In general, the torque T generated by a DC motor is proportional to the armature current i and the strength of the magnetic field. We assume the magnetic field to be constant and thus the torque is calculated as

$$T = K_t i, \quad (9)$$

where K_t is the motor torque constant. We denote the angular position of the motor to be θ . The angular velocity and acceleration are then $\dot{\theta}$ and $\ddot{\theta}$, respectively. The back EMF is proportional to the angular velocity of the shaft by a constant factor K_e as follows:

$$e = K_e \dot{\theta}. \quad (10)$$

A viscous friction model is assumed and the friction torque is proportional to the shaft angular velocity $\dot{\theta}$ by a factor of b . Now we can derive the following governing equations based on Newton's second law and Kirchhoff's law:

$$\begin{aligned} J\ddot{\theta} + b\dot{\theta} &= K_t i, \\ L \frac{di}{dt} + Ri &= V_{eff} - K_e \dot{\theta}, \end{aligned} \quad (11)$$

where J is the moment of inertia of the motor. It is noted that in the steady state (i.e., $\ddot{\theta} = 0$),

$$\dot{\theta} = \frac{K_t i}{b}. \quad (12)$$

The state-space system modelling as in (1) becomes

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} &= \begin{bmatrix} -\frac{b}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V_{eff}, \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}. \end{aligned} \quad (13)$$

The states are the angular velocity of the motor $\dot{\theta}$, constrained in $[0, \dot{\theta}_{\max}]$, and the armature current i , constrained in $[0, i_{\max}]$. The control input is the effective voltage V_{eff} , constrained in $[0, V]$ as discussed above. The system output is $\dot{\theta}$. The control goal is to make $\dot{\theta}$ track r .

2.2 Battery

Batteries are increasingly used as power source for many applications nowadays ranging from low-power applications such as portable electronics, wearable devices to high-power applications such as EVs and stationary electrical energy storage (EES) systems for smart grid applications [3]. Lithium-ion battery chemistry has been dominating the market for most low-power and high-power applications mainly due to their high energy and power densities compared to other rechargeable battery chemistry. While the terminal voltage and nominal capacity of a single

lithium-ion cell are limited for achieving high operating voltages and high capacities required for EVs, multiple individual lithium-ion cells are combined in series or parallel to form a high-power battery pack.

Major concerns affecting the widespread adoption of EVs include range anxiety and battery degradation that will result in an early replacement of their power source. For instance, battery packs in EVs have to be replaced when their state-of-health (SoH), a ratio of capacity at present to the capacity when the battery was new, falls below 70%. In addition to the long-term aging, battery packs are also subject to capacity degradation within individual charging–discharging cycles. This is mainly due to the *rate capacity* effect, which states that discharging a battery with a higher current will reduce the overall capacity of the pack that can be used in this cycle. Therefore, while designing control applications that use battery as a power source, the capacity degradation at single charging–discharging cycles and long-term battery aging have to be considered for maximizing the battery usage and its lifetime.

2.2.1 Battery Basics

Batteries are electrochemical storage devices, meaning their chemical reaction is coupled with an electron transfer. They perform a reversible chemical reaction, which allows them to store electrical energy (*charging*) and release the stored electrical energy by performing the opposite reaction (*discharging*). The basic unit of a rechargeable battery is an electrochemical cell, which consists of a positive electrode *cathode*, a negative electrode *anode*, and an electrolyte to favour the movement of the charge carriers between the two electrodes inside the cell as shown in Fig. 3.

Fig. 3 Electrochemical cell. Shuttle ions (M^+) are oxidized at the anode and move towards the cathode releasing an electron (e^-) to the outer circuit powering the load [13]

During the discharging process, shuttle ions (M^+) are oxidized at the anode side and release electrons (e^-), which travel through the outer circuit to power the load. The oxidized shuttle ions move through the electrolyte to the cathode inside the cell and are reduced by the incoming electrons from the outer circuit. This process is represented by the following equations:

The opposite reaction takes place during charging, facilitating storage of electrical energy in the form of chemical reactions.

In the ideal case, one would assume that while discharging the voltage of the electrochemical cell as seen by the load stays constant throughout the discharging process and suddenly drops to zero when the battery capacity is empty. Moreover, the capacity of the battery stays constant irrespective of the amplitude of the discharge current. However, in reality, the battery exhibits several non-linear effects and as a result the battery voltage instead of remaining constant slowly decreases with time while discharging. Furthermore, the usable capacity of a battery significantly depends on the rate of the load current. Discharging a battery with a higher current will result in a reduced effective capacity obtained from the cell.

2.2.2 Rate Capacity Effect

The FCC of a battery pack is reduced when a battery is discharged with a higher discharge current [5]. This can be seen from Fig. 4 where discharging a cell with a higher current reaches the lower threshold voltage faster than discharging with a lower current. This effect is termed as *rate capacity* or *rate effect*. The fundamental concept behind the rate capacity effect can be explained in terms of overpotential as in [13]. Whenever a current is drawn from a battery, the voltage of that battery will drop depending upon the magnitude of the discharging current. For a battery to obtain maximum energy output, the cell voltage V_T should follow the discharge profile of the equilibrium voltage V_0 , which is defined as the cell voltage at the chemical equilibrium at a given state of charge and temperature. However, the cell voltage deviates with the discharging current and this deviation is termed as overpotential η , which can be expressed as

$$\eta = V_0 - V_T. \quad (16)$$

This overpotential is mainly divided into three parts as ohmic, activation, and concentration overpotentials. At higher states of charge, the cell voltage is predominantly dominated by ohmic overpotentials, which behaves like a resistive drop to the cell voltage and as the cell discharges to a lower state of charge, the activation and

Fig. 4 Higher discharge current results in a reduced usable capacity of the cell due to rate capacity effect. The discharge rate is given by C rating, where 1C means the cell is discharged completely within 1 h [9]

concentration overpotentials dominate the ohmic drop. This reduction in cell voltage and the capacity due to the overpotentials of the cell is termed as rate capacity effect.

In battery terminology, the *C-rate* is often used to define the charge or discharge current of a battery. 1C corresponds to the current necessary to charge or discharge the battery completely in 1 h, whereas a 2C discharge will deplete the battery in half hour. The rate capacity effect is modelled by using Peukert's law [9] as

$$L = \frac{a}{I^b}, \quad (17)$$

where L is the battery lifetime, I is the discharge current, a and b are constants obtained from experiments. In ideal case a would be the battery capacity and b would be equal to 1, whereas in reality a is close to the battery's capacity and b is greater than one. While this model holds good for predicting battery capacity for constant continuous load, it does not work well with variable or interrupted loads. An extended version of Peukert's law was proposed in [15] as

$$L_t = \frac{a}{\left(\frac{\sum_{k=1}^n I_k (t'_{k+1} - t'_k)}{L_t} \right)^b}, \quad (18)$$

where $t'_1 = 0$ is the starting time stamp and $L_t = t'_{n+1}$ is the total duration that the battery can be used and divided into n slots.

2.2.3 Battery Aging

In addition to the single cycle capacity loss due to rate capacity effect, which can be rectified by reducing the discharging current at subsequent cycles, battery aging is a long-term process where the battery cell cannot hold the same amount of charge as it was new. Battery aging can be classified into calendar aging and cycling aging, where the former refers to the loss of capacity due to storing at high states of charge and high temperatures and the latter refers to the loss of lithium-ions due to the charge/discharge process. The main factors for battery cycling aging are depth of discharge (DoD), average state of charge, state-of-charge swing, temperature, and the rate of the discharge current [18] as

$$Q_{loss} = f(t, T, DoD, Rate), \quad (19)$$

where t is the cycling time. Without the DoD, which does not significantly affect the cycling capability of lithium-ion cells, the capacity loss can be modelled with the following equation:

$$Q_{loss} = B.exp\left(\frac{-E_a}{RT}\right)(A_h)^z, \quad (20)$$

where R is the gas constant, T is the temperature, A_h is the ampere-hour throughput of the cell, z is the power law factor, and B is a constant obtained by experimental data. With multiple experimental analysis for different discharge rates performed in [18], the value of z was approximated to 0.55 and the constant B was calculated for each C-rate. Figure 5 shows that with a higher discharge current the battery capacity drops significantly and will reach their end-of-life faster than discharging at a lower discharge current.

Fig. 5 Capacity loss due to higher discharge rates [5]

The capacity loss with discharging can be approximately modelled by the following equation as proposed in [18]:

$$Q_{loss} = B.exp \left[\frac{-31700 + 370.3 \cdot C_{rate}}{RT} \right] (A_h)^{0.55}. \quad (21)$$

2.3 Processor Aging

Processors are known to age over time and stress resulting in reduced operating speed. This is problematic in the sense that lower processor speed negatively impacts the performance of applications running on it.

2.3.1 Aging Mechanisms

The main transistor aging mechanisms are hot carrier injection and negative bias temperature instability [1, 17]. Hot carrier injection results in changes in the threshold voltage of the semiconductor. Similarly in the case of negative bias temperature instability, the threshold voltage of MOSFETs is increased. These aging-induced voltage changes result in longer transistor switching time. And as a consequence, the operation of the transistor becomes less reliable, which is of course highly undesirable. Such increased switching time lowers the performance of a processor and of applications running on the processor. Applications then potentially violate performance requirements and produce faulty calculation results, which in most cases is not acceptable.

2.3.2 Countermeasures

As a countermeasure to increased path delays, chips typically would run at very conservative clock rates, also called guard bands or safety margins. Such guard bands include enough margin to achieve the same clock rate throughout the whole intended lifetime of a processor. Intuitively, we see that this pessimistic approach results in a huge waste of resources or energy as the processor could generally achieve much higher speeds [11]. The problem even becomes more severe as we see a trend to decreasing transistor sizes which increases operational variations.

However an increase in the supply voltage could compensate for aging circuits [16]. By that, the delays could be kept constant and the operating frequency could stay the same throughout the intended lifetime of the processor. An adaptive control circuit accommodates for the currently required voltage settings. The downside of this approach is quadratically increased dynamic power consumption of the processor [14] and additional constraints such as maximum input current, cooling requirements, and temperature-dependent reliability problems [17].

Another measure to be taken is to decrease the operating frequency of the processor to compensate for critical path delays [1]. Other than increasing the supply voltage, decreasing the operating frequency actually lowers the power consumption. However, the processor becomes slower, which results in degraded control performance [4] and schedulability issues [12]. Nevertheless, dynamic operating frequency adjustments are a promising approach due to not negatively impacting the overall energy consumption while maintaining high usage of resources.

If aging could be measured on-chip in real-time, the operating voltage and frequency could be adjusted to always provide the maximum speed possible. This however means that control applications—that were designed for a higher speed, which becomes infeasible at some point in time—need to be readjusted to the changes. Such on-chip aging monitors have been developed for the delay of critical paths. Paths that potentially become critical in the future need to be identified first and then their degradation needs to be watched [19]. The path timing monitors typically work on replications of the paths that have statically been identified as the critical ones to not interfere with real functions. The information gathered from the replicated paths is then used to decide if the operating frequency needs to be changed and the according new frequency can be determined from the monitored delays. Processors that implement such critical path monitors are also called autonomic frequency scaling processors.

From the application designers' perspective, the processors lose speed over time and this needs to be considered when designing applications that will run on those processors. Lower processor operating frequency results in longer worst-case execution time of programs. However, in control applications that require high sampling frequencies this worst-case execution time may become the bottleneck for reliable control output. As already outlined above, safety margins between worst-case execution time and sampling period are possible but costly as the processor would run much below its capabilities throughout most of its lifetime. Hence, making full use of the available resources, here the processor speed, is of high interest in cost-sensitive domains.

2.3.3 Aging Estimation

A simple model of critical path delays uses temperature, supply voltage, and stress time, i.e., time during which the processor is active [12]. Let us use this model to consider the use case of processor aging in an electrical taxi. Assuming that the electrical taxi is in use for two-thirds of a day with drivers taking shifts, i.e., for 16 h, we can now estimate the decrease of the processor speed used in the car. We find that after 2 years of taxi use, the on-time of the processor is approximately 1.33 years. As a consequence, the processor speed has degraded in the worst case by roughly 7%. The degradation then continues. After four and then 10 years, the corresponding duration of the processor being switched on amounts to 2.66 and 6.66 years, respectively. These on-times relate to critical path delays of roughly 9% and 12%. As a vehicle usually incorporates many real-time and safety-critical tasks on

multiple processors and at the same time the automotive domain can be considered to be very cost-sensitive, such delays need to be considered in the design stage.

2.3.4 Related Work from the Software Perspective

Multiple works have proposed techniques to design software for aged processors or to reduce the aging process. Processors that have slowed down due to aging have higher execution delays of tasks, which is particularly problematic in the context of hard real-time systems and safety-critical applications [12]. As a result, the schedulability analysis for such safety-critical systems needs to take the estimated worst-case execution delays into consideration and the traditional problem formulation needs to be extended by system lifetime constraints. All scheduled tasks with worst-case delays need to meet their respective deadlines at all times even with severe aging-induced slow-down of the processor speed.

Mitigation of aging can be done in multi-core systems. Such systems often use redundant multi-threading to reduce soft errors. The aging variation among cores is due to varying workloads. Such unbalanced aging states are highly undesired as the system lifetime is constrained by the weakest component, i.e., the slowest core. As a remedy, the mapping of tasks should consider the current aging status of the respective cores. The proposed system [10] maps tasks in a way that aging variations are mitigated and aging of already slower cores is reduced.

3 Reliable CPS Design Framework

We formulate the reliable CPS design on unreliable hardware platforms to be an optimization problem with two objectives— t_s to quantify QoC and L_t to quantify the battery usage. We aim to minimize t_s and maximize L_t . Usually an optimization technique takes objectives either to minimize or maximize but not both. Therefore, we minimize $f_1 = t_s$ and $f_2 = -L_t$. It is noted that L_t is only related to the single-cycle behaviour with the battery rate capacity effect. Other objectives with respect to battery aging can also be defined, such as the total duration that a battery can run the control task after some time like 1 year, i.e., the L_t in 1 year, or when the capacity drops below the threshold like 70%.

The constraints are on the plant states and control input. Additional constraints on the objectives can be imposed depending on the requirements. For example, t_s can be set as shorter than or equal to 20 s. The decision variables are the poles that are less than unity on the real non-negative plane. Clearly, the decision space is continuous. Given a set of decision variables, the objectives and constraints can be evaluated as explained in Sect. 2.

There are generally two goals to pursue in solving bi-objective or multi-objective optimization problems. First, the final solution set (i.e., the obtained Pareto front) only consists of non-dominated points. By convention, Point A is said to dominate

Point B, if Point A is better than or equal to Point B in all objectives and better than Point B in at least one objective. Second, the final solution set has a good distribution in terms of objective values. This gives designers better options under different circumstances.

It is challenging to solve the formulated non-convex optimization problem with a continuous design space. Stochastic population-based heuristics such as the non-dominated sorting genetic algorithm (NSGA) can be used. In NSGA, an initial population is first generated and serves as parents. Offspring are then produced with crossover and mutation. The crossover function tries to keep the good genes of parents, which in this context means that the offspring are close to parents in the decision space.¹ The mutation function aims to better explore the decision space. Elitism is implemented for environmental selection, so that the next generation is selected among both the parents and offspring. This not only speeds up convergence but also ensures that good solutions will not be lost once they are found. There are two termination conditions whether the population has converged and whether the maximum allowed number of generations has been reached.

In selection, all the parents and offspring are sorted and ranked by domination. For each point, the number of points that dominate it (i.e., dominating points) is its rank. The new generation is filled in a way that points with lower ranks have priorities. This sorting feature values dominance more than the differences in individual objectives.

Among all the non-dominated points obtained by the above NSGA-based optimization, some may be very close to others in both the objectives. Therefore, it is not necessary to keep all of them. We need to choose a few points to form a well-distributed final solution set. First of all, we define the crowding distance below. As illustrated in Fig. 6, assuming that there are two objectives $\{f_1, f_2\}$ and n solution points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ² ordered by the value of either objective, for each point $\mathbf{x}_i, i \in \{1, 2, \dots, n\}$, that is not at the end of this point sequence, the crowding distance of $\mathbf{x}_i$ in terms of the objective $f_k, k \in \{1, 2\}$ can be calculated as

$$q_i^k = |f_k(\mathbf{x}_{i+1}) - f_k(\mathbf{x}_{i-1})|, \quad (22)$$

Fig. 6 Illustration of the crowding distance calculation

¹This may not be the case in general. Offspring can be quite different from the parents.

²These are just general notations to explain the method. In this chapter, the decision variables are the poles as discussed earlier.

Algorithm 1 Removal of less representative solution points according to the crowding distance ranking

Input: $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, n_d, \{\rho_1, \rho_2\}, \{f_1, f_2\}$

Output: $S_d = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_d}\}$

```

26 for  $j \in \{1, 2, \dots, n - n_d\}$  do
27   for  $i \in \{1, 2, \dots, n - j + 1\}$  do
28     calculate  $q_i^1$  and  $q_i^2$  for the element  $\mathbf{x}_i$  as in (22)
29   for  $k \in \{1, 2\}$  do
30     sort  $S$  based on  $q_i^k$  from maximum to minimum for  $i \in \{1, 2, \dots, n - j + 1\}$  do
31       assign the position index (1 for maximum to  $n - j + 1$  for minimum) of  $\mathbf{x}_i$  in  $S$  to
        $r_i^k$ 
32   for  $i \in \{1, 2, \dots, n - j + 1\}$  do
33     calculate  $r_i^0$  as in (23)
34   Remove the element with the maximum  $r_i^0$  from  $S$ 
35  $S_d = S$ 

```

where $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i-1}$ are the two closest points to $\mathbf{x}_i$ on each side, respectively. Since we deal with a set of Pareto points that are non-dominated, $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i-1}$ are closest to $\mathbf{x}_i$ in terms of both objectives. Both the end points of the point sequence are assumed to have infinite crowding distance calculation.

The algorithm removing the less representative points to achieve a good distribution is shown in Algorithm 1. The desired number of Pareto points is denoted as n_d . First, for each point, we calculate the two crowding distances corresponding to the two objectives. (Lines 2–4) Two ranks r^1 and r^2 are assigned to it based on the comparison in crowding distances with other points. (Lines 5–10) If the point $\mathbf{x}_i$ has the maximum crowding distance in terms of f_1 among all the n points, then $r_i^1 = 1$. If $\mathbf{x}_i$ has the minimum crowding distance, then $r_i^1 = n$. The overall rank of $\mathbf{x}_i$ is

$$r_i^0 = \rho_1 r_i^1 + \rho_2 r_i^2, \quad (23)$$

where ρ_1 and ρ_2 are importance factors of the two objectives, respectively. (Lines 11–13) These values depend on the application and

$$\rho_1 + \rho_2 = 1. \quad (24)$$

For example, if in an application, only the distribution in terms of f_1 is important, we may set ρ_1 to be 1 and ρ_2 to be 0. In this case, r_i^0 is equal to r_i^1 and all the points are ranked according to their crowding distances in terms of f_1 . After each point $\mathbf{x}_i$ has an overall rank r_i^0 , the point with the largest r_i^0 is removed from the solution set. (Line 14) The entire process starting from crowding distance calculation is iterated until the desired number of points n_d is reached. Both the end points of the point sequence are always kept in the set (due to the infinite crowding distances)

Fig. 7 An example trade-off between QoC and battery usage

Table 1 The ten design options: the original values and the aged values

Design option	t_s	L_t	Aged t_s		Aged L_t	
1	1.1257 s	4.2643 h	1.2369 s	9.88%	4.4601 h	4.59%
2	2.8972 s	5.4000 h	3.1859 s	9.96%	5.2717 h	−2.38%
3	3.9540 s	5.4182 h	4.3483 s	9.97%	5.2742 h	−2.66%
4	5.6430 s	5.5315 h	6.2062 s	9.98%	5.3111 h	−3.98%
5	7.2314 s	5.5412 h	7.9534 s	9.99%	5.5423 h	0.02%
6	8.2142 s	5.5959 h	9.0345 s	9.99%	5.5141 h	−1.46%
7	10.0238 s	5.6377 h	11.0251 s	9.99%	5.5639 h	−1.31%
8	11.1746 s	5.6506 h	12.2910 s	9.99%	5.5627 h	−1.56%
9	13.3920 s	5.6663 h	14.7302 s	9.99%	5.4613 h	−3.62%
10	18.0271 s	5.6930 h	19.8287 s	9.99%	5.5239 h	−2.97%

The percentage with the aged values is computed based on the original values

to maintain the coverage of the solution set. It is noted that Algorithm 1 takes two objectives into account and can be trivially extended for more objectives.

An example trade-off between QoC and battery usage with the electric motor control presented earlier in this chapter is illustrated in Fig. 7. As the processor ages, there is a decrease in the processor operating frequency and an increase in the sampling period. Taking the number 10% as an example, the change of both objectives is reported in Table 1. In 8 out of the 10 design options shown in Fig. 7, the aged points are dominated by the original points. That is, the settling time is increased (with a positive percentage) and the battery usage is decreased (with a negative percentage). The average deterioration in the control performance is 9.97%. The average deterioration in the battery usage is 1.53%. It is noted that for design option 2, the constraints on the plant states and control input, as discussed earlier in this chapter, are not satisfied anymore.

The processor aging effect can be mitigated by re-optimizing the controller poles with the prolonged sampling period, using the design framework earlier in this chapter. After obtaining the Pareto front, there can be different ways to reach the final solution set. For instance, Algorithm 1 can be deployed again. Alternatively, for each design option, we can keep the point that is closest to the original point

Table 2 The ten design options: the recovered values with re-optimization

Design option	Recovered t_s		Recovered L_t	
1	1.1475 s	−7.23%	4.2345 h	−5.06%
2	2.8389 s	−10.89%	5.3361 h	1.22%
3	3.9264 s	−9.70%	5.4116 h	2.60%
4	5.9551 s	−4.05%	5.5294 h	4.11%
5	7.5233 s	−5.41%	5.5676 h	0.46%
6	8.1094 s	−10.24%	5.5754 h	1.11%
7	10.8225 s	−1.84%	5.6383 h	1.34%
8	10.8225 s	−11.95%	5.6383 h	1.36%
9	13.7615 s	−6.58%	5.6595 h	3.63%
10	13.7615 s	−30.60%	5.6595 h	2.45%

The percentage is computed based on the aged values

in the settling time. The latter is executed in this case. The recovered results after re-optimization are shown in Table 2. In 9 out of the 10 design options, the recovered points dominate the aged points. That is, the settling time is decreased (with a negative percentage) and the battery usage is increased (with a positive percentage). The average improvement in the control performance is 9.85%. The average improvement in the battery usage is 1.32%. It should be noted that the design options 7 and 8 have the same recovered point. So do the design options 9 and 10. For all the design options, the constraints on the plant states and control input are guaranteed to be satisfied.

4 Concluding Remarks

In this chapter, we have discussed a design optimization framework for CPS. We consider unreliable hardware platforms with respect to processor aging and battery aging and rate capacity effect. The trade-off between the QoC and battery usage is explored. Furthermore, when the processor ages, both the QoC and battery usage get deteriorated, and safety requirements may be violated. The processor aging effect can be mitigated by re-optimizing the controller with the prolonged sampling period, using the same design framework. The change of QoC is minimal and the safety requirements are guaranteed to be met—leading to reliable CPS design. Besides the processor and battery, there are other hardware components that can be unreliable and should be investigated, e.g., the memory and communication systems [7].

References

1. Bowman, K., Tschanz, J., Wilkerson, C., Lu, S.L., Karnik, T., De, V., Borkar, S.: Circuit techniques for dynamic variation tolerance. In: 2009 46th ACM/IEEE Design Automation Conference, pp. 4–7. IEEE, New York (2009)
2. Chang, W., Chakraborty, S.: Resource-aware automotive control systems design: a cyber-physical systems approach. *Found. Trends Electron. Des. Autom.* **10**(4), 249–369 (2016)
3. Chang, W., Lukasiewicz, M., Steinhorst, S., Chakraborty, S.: Dimensioning and configuration of ees systems for electric vehicles with boundary-conditioned adaptive scalarization. In: International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) (2013)
4. Chang, W., Pröbstl, A., Goswami, D., Zamani, M., Chakraborty, S.: Battery-and aging-aware embedded control systems for electric vehicles. In: 2014 IEEE Real-Time Systems Symposium, pp. 238–248. IEEE, New York (2014)
5. Chang, W., Proebstl, A., Goswami, D., Zamani, M., Chakraborty, S.: Reliable CPS design for mitigating semiconductor and battery aging in electric vehicles. In: 2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications, pp. 37–42 (2015)
6. Chang, W., Roy, D., Zhang, L., Chakraborty, S.: Model-based design of resource-efficient automotive control software. In: IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (2016)
7. Chang, W., Goswami, D., Chakraborty, S., Ju, L., Xue, C., Andalam, S.: Memory-aware embedded control systems design. *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.* **36**(4), 586–599 (2017)
8. Chang, W., Goswami, D., Chakraborty, S., Hamann, A.: OS-aware automotive controller design using non-uniform sampling. *ACM Trans. Cyber-Phys. Syst.* **2**(4), 26 (2018)
9. Jongerden, M., Haverkort, B.: Battery Modeling. No. TR-CTIT-08-01 in CTIT Technical Report Series, Design and Analysis of Communication Systems (DACS) (2008)
10. Knebel, F., Rehman, S., Shafique, M., Henkel, J.: Ageopt-rmt: compiler-driven variation-aware aging optimization for redundant multithreading. In: 2016 53rd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2016). doi:[10.1145/2897937.2897980](https://doi.org/10.1145/2897937.2897980)
11. Lefurgy, C.R., Drake, A.J., Floyd, M.S., Allen-Ware, M.S., Brock, B., Tierno, J.A., Carter, J.B.: Active management of timing guardband to save energy in power7. In: Proceedings of the 2011 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1–11. IEEE, New York (2011)
12. Masrur, A., Kindt, P., Becker, M., Chakraborty, S., Kleeberger, V., Barke, M., Schlichtmann, U.: Schedulability analysis for processors with aging-aware autonomic frequency scaling. In: Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 11–20. IEEE, New York (2012)
13. Narayanaswamy, S., Schlueter, S., Steinhorst, S., Lukasiewicz, M., Chakraborty, S., Hoster, H.E.: On battery recovery effect in wireless sensor nodes. *ACM Trans. Des. Autom. Electron. Syst.* **21**(4), 60:1–60:28 (2016)
14. Park, J., Abraham, J.A.: A fast, accurate and simple critical path monitor for improving energy-delay product in dvs systems. In: Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design, pp. 391–396. IEEE, New York (2011)
15. Rakhmatov, D.N., Vrudhula, S.B.K.: An analytical high-level battery model for use in energy management of portable electronic systems. In: IEEE/ACM International Conference on Computer Aided Design (ICCAD 2001). IEEE/ACM Digest of Technical Papers (Cat. No.01CH37281), pp. 488–493 (2001)
16. Stojanovic, V., Markovic, D., Nikolic, B., Horowitz, M.A., Brodersen, R.W.: Energy-delay tradeoffs in combinational logic using gate sizing and supply voltage optimization. In: Proceedings of the 28th European Solid-State Circuits Conference, pp. 211–214. IEEE, New York (2002)

17. Tschanz, J., Kim, N.S., Dighe, S., Howard, J., Ruhl, G., Vangal, S., Narendra, S., Hoskote, Y., Wilson, H., Lam, C., et al.: Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging. In: 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, pp. 292–604. IEEE, New York
18. Wang, J., Liu, P., Hicks-Garner, J., Sherman, E., Soukiazian, S., Verbrugge, M., Tataria, H., Musser, J., Finamore, P.: Cycle-life model for graphite-lifepo4 cells. *J. Power Sour.* **196**(8), 3942–3948 (2011)
19. Wang, S., Chen, J., Tehranipoor, M.: Representative critical reliability paths for low-cost and accurate on-chip aging evaluation. In: Proceedings of the International Conference on Computer-Aided Design (ICCAD '12), pp. 736–741. ACM, New York (2012). doi:10.1145/2429384.2429543. <http://doi.acm.org/10.1145/2429384.2429543>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.