May 26, 2021

Transformation of controller software for ensuring safe behavior under perceptual uncertainty

-- Bridging the gap between uncertain perception and reality in controller software design --

A research team consisting of Tsutomu Kobayashi, Ichiro Hasuo, Fuyuki Ishikawa, and Shin-ya Katsumata at the National Institute of Informatics (NII, Japan) and Rick Salay and Krzysztof Czarnecki at University of Waterloo (Canada) developed a method that automatically transforms models of controller software into models that satisfy safety requirements even when there is uncertainty in sensing the state of the environment. In addition to the transformation, the method generates formulas that represent the degree of uncertainty that the controller software can tolerate. The method can be applied to various controller systems that interact with the external environment, including autonomous vehicles.

This research was conducted under the ERATO MMSD Project [*1] funded by the Japan Science and Technology Agency (JST, Japan). The findings were presented at the 13th NASA Formal Methods Symposium (online) on May 26th, 2021.

Background

Controller systems such as drones and autonomous vehicles are vital in society. These systems must be able to function as safely as possible because many of them are intended for use in the real world. Approaches involving mathematical modeling of the systems and verifying their safety are effective for guaranteeing safety.

The controller software in such systems determines its actions in response to the state of the environment, which is perceived with sensors. In reality, however, the systems may perceive values that differ from the true values (perceptual uncertainty). This can cause safety violations if the controller software behaves in response to the incorrectly perceived values. For instance, if the sensor of an autonomous vehicle can misperceive the positions of other cars up to 1 m, it should operate with a safety margin of at least 1 m (Figure 1 (a)).

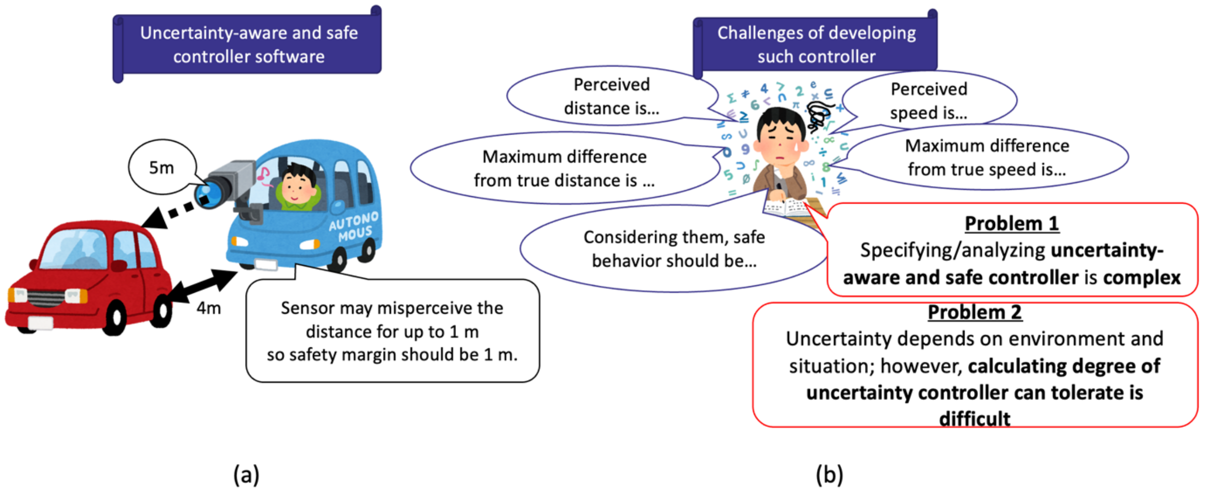
Designing such uncertainty-aware and safe controller software is quite complicated because developers need to verify that safety is guaranteed for every possible behavior of the system while taking into consideration differences between true values and perceived values (Figure 1 (b)). In addition, it is difficult to estimate the degree of the uncertainty. For example, perceptual uncertainty depends on the situation in which the controller system is deployed, such as whether or not it is foggy. Therefore, rather

Japan Science and Technology Agency（JST）

Public Relation Division
5-3 Yonban-cho Chiyoda-ku, Tokyo
102-8666 JAPAN
TEL：+81(0)3-5214-8404　FAX：+81(0)3-5214-8432
E-Mail：jstkoho@jst.go.jp

Research Organization of Information and Systems
National Institute of Informatics
Publicity Team
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo
101-8430 JAPAN
Direct: +81(0)3-4212-2164　FAX：+81(0)3-4212-2150
E-Mail: media@nii.ac.jp

than specifying concrete perceptual uncertainty in models, constructing uncertainty-unaware controller models and calculating the degree of uncertainty the constructed controllers can tolerate are more suitable for flexible analysis. However, obtaining this limit as a formula proved to be difficult.

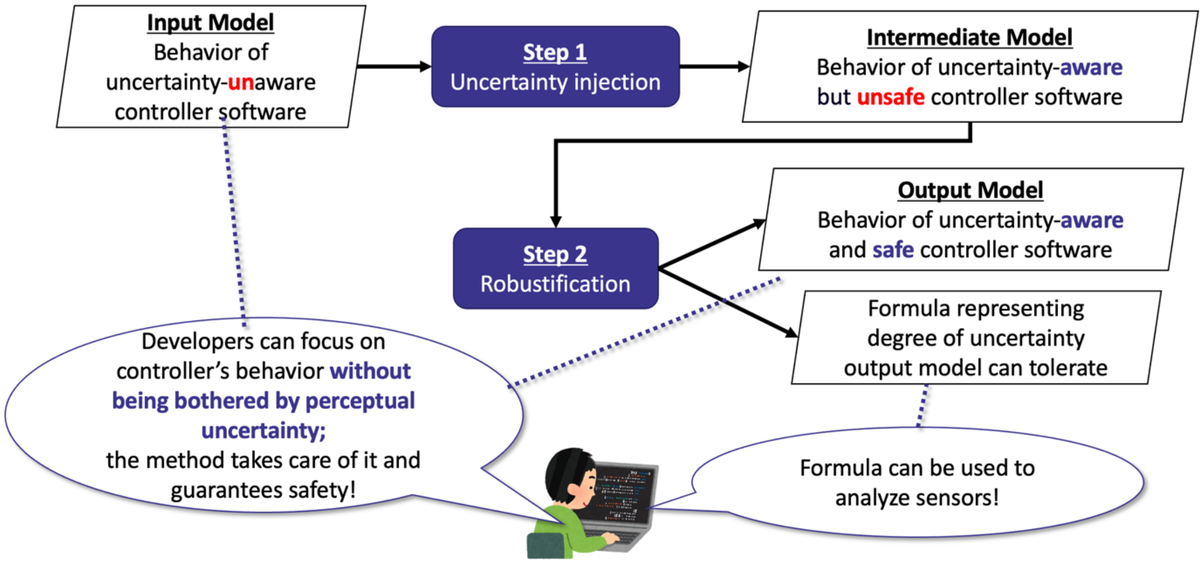


Uncertainty-aware and safe controller software
5m
4m
AUTONOMOUS
Sensor may misperceive the distance for up to 1 m so safety margin should be 1 m.
Challenges of developing such controller
Perceived distance is...
Maximum difference from true distance is ...
Perceived speed is...
Maximum difference from true speed is...
Considering them, safe behavior should be...
Problem 1
Specifying/analyzing uncertainty-aware and safe controller is complex
Problem 2
Uncertainty depends on environment and situation; however, calculating degree of uncertainty controller can tolerate is difficult
(a)
(b)

Figure 1. (a) Example of safe uncertainty-aware controller software (b) Challenges of developing such controller software equipped with uncertain sensors.

Methods and Results

In this study, we propose a method that automatically transforms a model of an uncertainty-unaware controller into a model of a robustified controller, that is, a new controller that safely behaves even under uncertainty (Figure 2).



Input Model
Behavior of uncertainty-unaware controller software
Step 1
Uncertainty injection
Intermediate Model
Behavior of uncertainty-aware but unsafe controller software
Step 2
Robustification
Output Model
Behavior of uncertainty-aware and safe controller software
Formula representing degree of uncertainty output model can tolerate
Developers can focus on controller's behavior without being bothered by perceptual uncertainty; the method takes care of it and guarantees safety!
Formula can be used to analyze sensors!

Figure 2. Overview of proposed method. An input model is transformed in two steps to generate a model of the robustified controller and calculate the degree of uncertainty that the generated controller can tolerate.

The method consists of two steps.

The first step (uncertainty injection) is to transform an input model of an uncertainty-unaware controller into an intermediate model of an uncertainty-aware controller. Note that the behavior of the intermediate model is the same as that of the input model so it is unsafe.

Then the next step (robustification) is to transform the intermediate model into one that is uncertainty-aware and safe. The behavior of the resulting controller is updated so that it operates safely even under uncertainty.

In addition, the method generates a formula that represents the degree of uncertainty that the output controller can tolerate.

In the first step (uncertainty injection), variables of perceived values, which can differ from true values, are introduced. Moreover, while the input model is specified under an ideal assumption that the controller can refer to the true value when it determines an action, the intermediate controller refers to the perceived value to reflect the reality of the perception. Although the intermediate controller determines its actions in response to perceived values, safety is not guaranteed because its behavior is the same as that of the input controller. For example, an autonomous vehicle may misperceive the distance to another car as 5 m when it is only 4 m, resulting in a collision.

In the second step (robustification), the case distinction of the state of the environment is calculated. For instance, in some situations, an autonomous vehicle may be unsure of whether it should cruise or brake due to perceptual uncertainty about the distance between itself and a car ahead. The robustification method exhaustively lists such uncertain cases so that they are considered separately. In addition, the behavior of the controller is updated so that safety will be guaranteed even under perceptual uncertainty. The method considers all possibilities of the true value estimated from the perceived value and calculates behavior that is safe for every possibility.

Constraints on the behavior of the generated controller are specified so that it is guaranteed to operate safely even under uncertainty. However, whether such constraints are satisfiable depends on the uncertainty. As an extreme example, if a sensor misperceives the positions of other cars up to 100 km, then guaranteeing safety is impossible in many situations. This raises the question of how much uncertainty the generated controller can tolerate. Where is the limit?

To answer this, the proposed approach also generates the limit as a formula of uncertainty. Developers can choose appropriate sensors from a given catalog by using the formula as the criterion. In addition, the formula can be used to analyze uncertainty, such as how the uncertainty will be propagated if the controller is combined with other components.

The method makes the construction of uncertainty-aware and safe controllers more systematic and effortless. Moreover, it enables developers to flexibly analyze various situations of perceptual uncertainty. Thus, the method improves the overall safety of the real world in which controller systems are implemented ubiquitously.

Future outlook

In addition to autonomous vehicles, the proposed method can also be applied to various other controller systems that interact with external environments. In future work, we intend to generalize the method so that it will be able to deal with a broader range of uncertainty. For instance, we will tackle misclassification problems such as an object classifier module classifying an object as a wrong object class.

Comments from Tsutomu Kobayashi

“Controller systems are crucial because most software systems' usefulness is due to their interactions with external environments. This research aims to help developers apply formal modeling approaches to realistic software by addressing the inevitable problem of controller systems regarding the gap between the perception and reality. Thus, developers can focus on the essence of controller behavior. We believe that the method is valuable and can be extended in various ways. We will continue working towards the systematic and easy application of rigorous mathematical methods to ensure a safe environment for everyone.”

Funding

This research has been made possible thanks to the support of the Japan Science and Technology Agency’s ERATO-MMSD (JPMJER1603).

Information on Paper

Title: Robustifying Controller Specifications of Cyber-Physical Systems Against Perceptual Uncertainty
Authors: Tsutomu Kobayashi, Rick Salay, Ichiro Hasuo, Krzysztof Czarnecki, Fuyuki Ishikawa, Shin-ya Katsumata
Conference: The 13th NASA Formal Methods Symposium https://shemesh.larc.nasa.gov/nfm2021/
Date of Presentation: May 26th, 2021 (US Eastern Time)

END

<Media Contact>
Research Organization of Information and Systems
National Institute for Informatics
Publicity Team
TEL: +81(0)3-4212-2164; E-mail: media@nii.ac.jp

Japan Science and Technology Agency (JST)
Public Relations Division
TEL：+81(0)3-5214-8404　E-mail：jstkoho@jst.go.jp

〈About JST〉
Japan Science and Technology Agency (JST)
Department of Research Project　Nobuhiro Uchida
TEL：+81(0)3-3512-3528　E-mail：eratowww@jst.go.jp

[*1] ERATO Hasuo Metamathematics for Systems Design Project (ERATO-MMSD): a project funded in the Exploratory Research for Advanced Technology (ERATO) scheme of the Japan Science and Technology Agency (JST). The project conducts academic research for quality assurance of cyber-physical systems as the core of Society 5.0. The project specifically focuses on automated driving systems and investigates reliability techniques for modeling, formal verification, testing, and holistic, practical V&V techniques including all of them. This challenge requires tight collaboration of different academic areas such as software science and engineering, control theory and engineering, and artificial intelligence. Therefore, the project also focuses on (meta)mathematical theories. https://www.jst.go.jp/erato/hasuo/en/