Technique finds software bugs in surgical robots, helps ensure safety

Carnegie Mellon

PITTSBURGH-Surgical robots could make some types of surgery safer and more effective, but proving that the software controlling these machines works as intended is problematic. Researchers at Carnegie Mellon University and the Johns Hopkins University Applied Physics Laboratory have demonstrated that methods for reliably detecting software bugs and ultimately verifying software safety can be applied successfully to this breed of robot.

They used theorem-proving techniques to analyze a control algorithm for a research robot that would help a surgeon perform surgery at the base of the skull. Their method identified a safety flaw that could enable a scalpel or other surgical tool to go dangerously astray in this area, where the eye orbits, ear canals and major arteries and nerves are closely spaced and vulnerable to injury. It also guided development of a new algorithm and verified that the new controller was safe and reliable.

"These techniques are going to change how people build robotic surgery systems," predicted APL's Yanni Kouskoulas, who led the research study with André Platzer, assistant professor of computer science at Carnegie Mellon. Platzer and Kouskoulas say this formal verification technique also could change the way regulators evaluate new devices, providing more assurance of safety than is possible even with the most rigorous testing.

The researchers will present their findings April 11 at HSCC 2013, the Hybrid Systems: Computation and Control conference in Philadelphia. Other members of the study team were David Renshaw, a student in computer science at Carnegie Mellon, and Peter Kazanzides, associate research professor of computer science at Johns Hopkins.

Surgical robots are an example of a hybrid, or cyber-physical system - complex, computer-controlled devices that are becoming increasingly common. Other examples are aircraft collision avoidance systems, high-speed train controls and cars that avoid collisions, maintain their lanes or otherwise drive themselves.

"Because the consequences of these systems malfunctioning are so great, finding a way to prove they are free of design errors has been one of the most important and pressing challenges in computer science," Platzer said. Testing alone is inadequate because no test regimen can check all of the possible circumstances that the system might encounter.

A growing number of techniques have been developed to aid in formally verifying that computer hardware and software are free from design defects. These

techniques analyze all the possible states of a system, much as a mathematician uses a proof to determine that a theorem is correct. But methods that work for computer circuitry or software, which may be complex but have a finite number of states, don't work for hybrid systems that must contend with the infinite variations of the physical world.

Platzer, however, has developed an approach based on differential dynamic logic and an associated tool called KeYmaeraD that can be used to model a hybrid system and its properties and then symbolically pick it apart. This approach, which Platzer already has used successfully to identify errors in aircraft collision avoidance systems and to verify the design of distributed car control systems, can verify that a design is safe or else help generate counterexamples of how the system can fail.

Platzer and his colleagues applied this approach to evaluate the control algorithm for the skull-base surgery robot. This robot aids in intricate surgery in small recesses of the brain by minimizing tiny movements as a surgeon manipulates a tool and by restricting the tool to movement within the surgical site. As the tool approaches the surgical boundary, beyond which healthy and vital tissues can be harmed, it exerts force feedback to warn the surgeon. If the tool reaches the boundary, the robot is supposed to stop it from going farther. This functionality is helpful for the surgeon, because the robot knows the delicate boundaries that the surgeon cannot necessarily see during the surgery.

Kouskoulas said the robot and the control algorithm were tested extensively, including on cadavers. "While it worked in the configurations in which it was tested, the fear was always that something unexpected could go wrong," he noted.

By using the formal verification method, the researchers showed that indeed something unexpected could occur in corners of the surgical site. They found that in some geometrical configurations, the safety feedback for one boundary would interfere with that of the adjoining boundary, canceling each other out and allowing the tool to be pushed beyond the limits set by the surgeon.

The tool generated examples of how this could occur. "It leads you to the problem," Kouskoulas explained. "You then have to be creative to find the solution." With that guidance, researchers were able to devise a new algorithm and use their method to prove it was safe.

"Medical robotics is an interesting problem area for hybrid systems," Platzer said. Existing certification procedures, which rely on trial-and-error testing, aren't appropriate for evaluating these software-intensive devices, he said. This study shows that formal verification methods can be applied successfully to medical robotics and that further development is warranted, he added.

Source:
http://www.cmu.edu/news/stories/archives/2013/april/april8_surgicalrobotbugfixes.html [1]

Source URL (retrieved on *03/11/2014 - 12:21pm*):
http://www.ecnmag.com/news/2013/04/technique-finds-software-bugs-surgical-robots-helps-ensure-safety

Links:
[1] http://www.cmu.edu/news/stories/archives/2013/april/april8_surgicalrobotbugfixes.html