2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Safety Design View: A Conceptual Framework for Systematic Understanding of Safety Features of Medical Robot Systems

Min Yang Jung, Russell H. Taylor, and Peter Kazanzides

Abstract— **A variety of medical and surgical robot systems have been developed in academia and industry and commercial products are actively used in modern operating rooms. However, there is no safety standard that specifically governs the design of medical robot systems. Despite the availability of several safety design guidelines, the absence of a basis or foundation for safety makes it difficult to describe safety designs in a systematic manner, and to share knowledge and experiences on safety with others. In the meantime, the scale and complexity of recent medical robot systems have been increasing and this further complicates the effective representation and sharing of safety designs. As an approach to this issue, we propose the Safety Design View, a conceptual framework that can capture and describe both the design-time and run-time characteristics of safety features of medical robot systems in a systematic and structured manner. To illustrate the application of the Safety Design View, we collected a set of frequently used safety features, based on our literature review of safety in the medical robotics domain, and show how we can more effectively describe and understand safety designs of medical robot systems.**

I. Introduction

Medical robots are examples of safety-critical systems because human lives depend on their correct operation. Medical robot safety is distinctly different from other applications of robotics because: (1) humans must be present in the robot's workspace, (2) one of the humans is usually anesthetized and cannot escape, and (3) the robot may be holding a sharp instrument and be required to "injure" the human to perform the surgical intervention. Despite these inherent safety concerns, medical robotics is an active and rapidly growing field because robots can enable less invasive and more accurate surgeries, to the benefit of patients.

Currently, there is no safety standard that specifically governs the design of medical robot systems; rather, developers conform to existing medical device standards, such as IEC-60601 and IEC-62304. Similarly, medical robots are subject to the same regulatory approval processes as other medical devices. As in other safety-critical domains, developers must invest significant engineering effort to ensure that every device they design is safe and meets regulatory requirements. This obviously leads to a desire to capture the "best practices" that can be reused when designing new systems.

One effective approach to capturing such practices is to establish a common basis or conceptual framework that can provide consistent and systematic views of safety designs, and thus allow us to systematically describe, understand, and share safety features or designs. In the domain, such a common ground is necessary for effectively exchanging knowledge and experiences on safety designs with the community. However, the majority of the prior works on safety of medical robot systems have been system-specific or application-specific, and there has not been much discussion about what constitutes safety features, what should be considered when designing safety features, and how they can be better organized and presented in a systematic way.

This work proposes a conceptual framework that identifies essential elements or enabling components of safety features of medical robot systems with consideration of run-time aspects of the systems. The starting point is to recognize safety as a system property [1]. We treat safety as an *emergent property* that has meaning only when considered at the system-level, not at the individual component level [2]. We also take into account issues related to the deployment of safety features, which could practically have a significant impact on their run-time characteristics or performance. These considerations lead us to define the *two views* of safety features: the *mechanism view* and the *system view* (Sec. III). Based on these two views, we define the *design space* of safety features, which can simultaneously present the mechanisms and run-time aspects of safety features (Sec. IV).

The conceptual framework is based on many years of experience building and observing medical robot systems both in industry and academia, as well as a review of the medical robotics literature (Sec. IV-B). With the proposed framework, our goal is to: (1) systematically understand the design and characteristics of safety features, (2) enable the accumulation of prior experiences in a structured manner, and (3) facilitate sharing of knowledge and experience on safety within the community. The remainder of the paper discusses the benefits, limitations, and opportunities for further improvement of the proposed views (Sec. V), as well as future works (Sec. VI).

II. Related Works

Safety has been studied in a wide variety of domains, both in industry and academia. This includes traditional safety-critical systems, such as automotive systems, aerospace systems, medical devices, nuclear power plants, chemical processing plants, railway control systems, and weapon control systems. Our focus is primarily on prior work done within the medical robotics domain. In this domain, although safety has been unanimously recognized as one of the most important properties of the system, we have not found prior works that are directly related to our efforts. However, there exists a body of work that is conceptually relevant: a) safety

Authors are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA. Peter Kazanzides can be contacted at pkaz@jhu.edu.

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

guidelines or classifications, and b) system architectures with consideration of safety.

Early works on safety guidelines or classifications started to appear in the early 1990s. Taylor [3] reported the safety design and implementation of an orthopaedic surgery robot of which safety mechanisms are based on consistency checking. Davies [4] presented a set of safety design guidelines with safety requirements. More recently, O'Toole [5] summarized design methods for safety of surgical robotics systems, and Dombre [6] presented design guidelines that classify safety design methods depending on the type of components (electromechanical, electrical, and software). Our prior work [7] has a preliminary classification of safety features as well.

As for the architecture of medical and surgical robot systems, a layered or hierarchical architecture has been widely adopted because it helps to reduce the system complexity by splitting concerns and objectives of the whole system into multiple layers. Such a layered or hierarchical architecture is found in many projects (e.g., [8], [9], [10]), including commercial surgical robot systems (ROBODOC [11] and da Vinci [12]). This architecture can be also found in the general robotics domain [13], [14].

In the software engineering domain, the 4+1 view model for describing the architecture of software-intensive systems was proposed [15]. Its concept and approach is similar to our work in that the model provides different perspectives on the system architecture, thereby facilitating systematic understanding and structured presentation of the system architecture.

III. Two Views of Safety Features

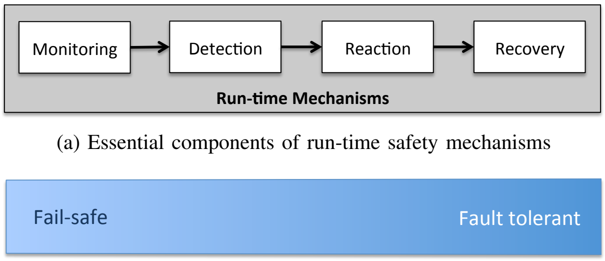
When designing safety features for a medical robot system, we consider two different aspects: *functional components* and *deployment options*. The functional components describe what a safety feature does and how it improves the safety of the system, and are usually derived from a set of safety requirements. The deployment options are about how to actually implement safety features and how and where to deploy them in the system. The design of safety features should take into account both aspects together because they are not independent of each other. Thus, we propose two views that reflect each aspect: the *mechanism view* and the *system view*. These two views enable structured descriptions of safety features with consideration of run-time characteristics.

In this section, we describe the definition and characteristics of each view, together with an example that illustrates how each view is applied to some representative safety features that have been frequently used in the medical robotics domain.

A. Mechanism View

The mechanism view (Fig. 1) defines *functional components* of safety features, and identifies the objective or behavior of a safety feature. Essentially, this view *decomposes* safety features into functional components.

1) Definition: The *run-time mechanisms* decompose safety features into four functionally essential components that provide a basis for understanding and presenting safety features. The four essential components are *monitoring*, *detection*, *reaction*, and *recovery*. These four components are based on our experience and observation in the domain, but similar concepts or models can also be found in other domains such as control systems [16].



(a) Essential components of run-time safety mechanisms

(b) Characteristics: Fail-safe systems focus on monitoring and detection, and fault tolerant systems put more emphasis on reaction and recovery.

Fig. 1: Mechanism View: Four essential components of run-time safety mechanisms with their characteristics

Monitoring (*"What to and how to monitor?"*): The monitoring mechanism continuously reads quantities or states of interest from the system. It is the starting point of run-time safety because it makes run-time data available to the system and allows the system, including the human operator, to be aware of its current status and the surrounding environment. One important requirement of the monitoring mechanism is its minimal run-time overhead on the target object being monitored. Otherwise, the monitoring mechanism may introduce adverse run-time impact or burden (i.e., performance degradation) on the target.

Detection (*"How to detect events?"*): The detection mechanism determines whether any event happened based on event specifications, and if it happened, it may include a process or subsystem that identifies detailed information about the event, such as severity, location, or timing. When designing and implementing a detection mechanism, it is crucial to minimize latency between the time when an event *actually* happened and the time when it is *determined* to have happened.

Reaction (*"What to do when events are detected?"*): The reaction mechanism defines initial and immediate responses to any erroneous or undesired event. Widely used methods in the domain are the *fail-safe* emergency pause (E-pause), which stops robot motion, or emergency stop (E-stop), which disables robot motor power. Both methods subsequently generate and propagate emergency events to the rest of the system. This approach has been accepted as a working solution, mainly due to the characteristic of the medical intervention where a fail-safe system is often sufficient [17].

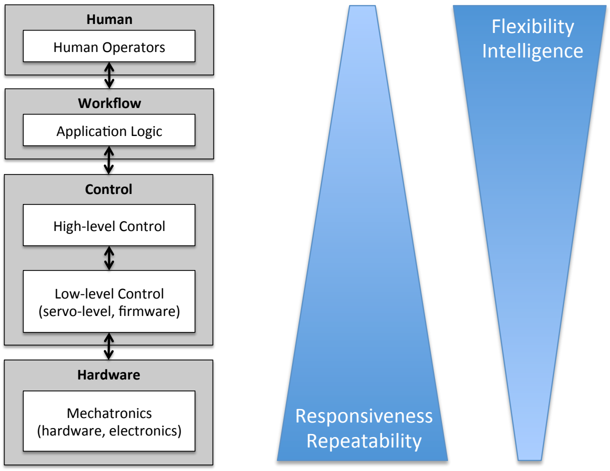
Recovery (*"How to recover from events?"*): The recovery mechanism represents policies, strategies, or methods to recover from erroneous states of a system. This mechanism may depend on human operators (e.g., workflow modifications such as switching to a conventional surgery when the robot fails), or automatically restore its normal state if the system is able to handle or tolerate such undesired events.

2) Characteristics: Fig. 1b shows characteristics of safety features that the mechanism view can reveal. A safety feature is considered to be *fail-safe* if the focus is on the monitoring and detection mechanism, and to be *fault tolerant* if more emphasis is placed on the reaction and recovery mechanisms. This distinction is not mutually exclusive. Rather, it represents the relative degree with which a safety feature focuses on the mechanisms. Kazanzides previously pointed out the differences between fail-safe and fault tolerant systems [17], noting that a fail-safe system is often sufficient for the medical intervention, but fault tolerance may be required for more advanced surgeries. One interpretation of this statement from the mechanism view's standpoint is that safety features of advanced medical robot systems should have better support for the reaction and recovery mechanisms.

3) Example: The use of a force sensor, and force threshold check, is one of the most widely used safety features in the medical robotics domain (see Sec. IV-B). A force sensor software module (e.g., an object or component) periodically *monitors* force feedback from the environment. If the module *detects* excessive force beyond a predefined threshold, it initiates the emergency pause or stop as an immediate *reaction*, which then stops or powers off the robot as quickly as possible. In the meantime, a signal is generated to inform the system of the event so that other parts of the system can take appropriate *reactions* (e.g., emergency alert for users, transition to E-pause or E-stop state). When the cause of the excessive force event is removed, the system can *recover* from the emergency state to continue its previous task.

B. System View

The system view presents a hierarchical structure of medical robot systems, as shown in Fig. 2. This hierarchical structure forms a layered architecture that has been widely used both in the robotics and the medical robotics domains [11], [13], [8], [14].


Human
Human Operators
Workflow
Application Logic
Control
High-level Control
Low-level Control (servo-level, firmware)
Hardware
Mechatronics (hardware, electronics)
Flexibility Intelligence
Responsiveness Repeatability

(a) Layered architecture of medical robot systems

(b) Characteristics of each layer

Fig. 2: System View: Layered architecture of medical robot systems with characteristics of each layer

1) Definition: The system view captures design decisions on the deployment of safety features by identifying which layer functional components of the safety feature are actually implemented. Those design decisions are important factors that determine the effectiveness of the safety features, and eventually the safety of the system. The system view should be tailored to reflect the actual system architecture. In this paper, we illustrate the concept using a canonical robot system architecture with the four groups of layers as in Fig. 2a: *Hardware*, *Control*, *Workflow*, and *Human*.

Hardware: The hardware layer represents mechanical and electrical components of a safety feature as well as the use of, or reliance on, physical devices such as various sensors (e.g., force/torque sensor, accelerometer) and external tracking devices (e.g., optical tracker).

Control: The control layer implements the control loop between the hardware and applications and provides a set of *application-independent*, but robot-specific, services for the upper layers (the *workflow* and *human* layers). This layer consists of two sub-layers: **high-level control** and **low-level control**. The high-level control performs tasks such as motion or trajectory planning and typically runs at hundreds of Hz. The low-level control refers to the servo-level control and is often implemented on dedicated devices (e.g., firmware on controller boards).

Workflow: The workflow layer implements *application-specific* logic or data (e.g., surgical planning, patient-specific data processing) on top of the services that the control layer provides. When developing multiple surgical scenarios or procedures that use the same robot, the separation of the workflow layer from the lower layers (hardware and control) facilitates the development process by enabling the reuse of resources and services from the control layer.

Human: The human layer represents activities or interactions with the human, such as human intervention (e.g., decision making or supervision). In this layer, the human means people that use the system (e.g., surgeons, system operators, medical personnel) and does not include patients.

2) Characteristics: Each layer of the system view has its own characteristics that have a profound influence on the characteristics of the run-time behavior of the safety features. We consider four characteristics of each layer, as in Fig. 2b.

Responsiveness: The hardware layer uses dedicated hardware and electronic circuits that are optimized for specific requirements. In contrast, humans have inherent physiological limitations in sensing and processing of sensory information (e.g., limits on the temporal resolution for visual stimuli, bandwidth limits on the range of audible sound). This makes the hardware layer the most responsive layer, and the human layer the least responsive one.

Repeatability: Repeatability is one important property of safety features because extensive and repetitive testings can prove that the safety features work as designed, and thus meet their safety requirements. Like responsiveness, the human layer has the least repeatable characteristic due to physiological fatigue, whereas the other layers can be heavily and thoroughly tested via automated unit-testing frameworks.

Flexibility: Flexibility can be considered from two respects: design flexibility (e.g., how easily can we change parameters or behaviors?) and adaptation flexibility (e.g., how well can a layer adapt to changes in environment or conditions?). For both cases, hardware provides the least flexible options to update parameters or to change logic, whereas humans can adapt to changes in the surrounding environment. In this sense, the hardware layer is least flexible and the human layer is most flexible.

Intelligence: The hardware layer has very specialized and limited "intelligence" (e.g., sensors, electronic elements, firmware) and thus can only handle changes of the surrounding environment that were anticipated during design. Humans, however, have experiences and expertise that can deal with unexpected events, and thus some safety features should rely on a human's decisions or supervision.

3) Example: We consider the force sensor-based safety feature that was used for the mechanism view example. This feature can be deployed to the system in different ways, depending on the vertical distribution of each runtime mechanism. For example, we can deploy all runtime mechanisms (i.e., monitoring, detection, reaction and recovery) to the high-level control. The implication of this option is that this design does not rely on a human's decisions, and the safety checking would be done as part of the control loop. Another option is to deploy the first three mechanisms to the high-level control layer and to rely on the human for the recovery mechanism. In this case, the recovery mechanism becomes less responsive, harder to test, but more adaptable to changes and can take advantage of the human's experience and intelligence.

IV. Safety Design View

The two views defined in the previous section allow us to look at two different aspects of safety features separately. Now we combine these two views into a two-dimensional plane, as shown in Fig. 3, with the following axes: (1) the mechanism view as the horizontal axis, and (2) the system view as the vertical axis. This two-dimensional plane forms the design space of safety features and we call it the Safety Design View (SDV).

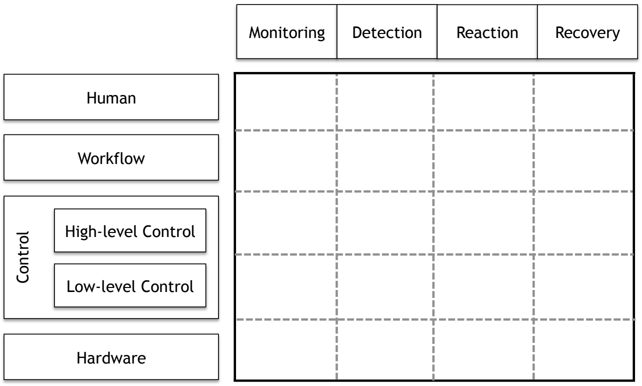


Fig. 3: Safety Design View: design space of safety features

A. Definition and Characteristics

SDV is a domain-specific view to elucidate the characteristics of safety features of medical robot systems in a consistent and systematic manner. SDV represents the design space of safety features as in Fig. 3 and we use the term "SDV" and the "design space" interchangeably in this paper.

SDV coherently presents the two different aspects of safety features–functional components and deployment decisions–based on the mechanism view and the system view. The characteristics of the two views also apply to SDV in the orthogonal directions to each axis. For example, characteristics of the high-level control layer in the vertical axis apply to the entire row of the layer, and the design requirements or issues with the monitoring mechanism apply to the entire column. This coherent presentation of safety features in the design space allows SDV to capture design decisions on how to deploy safety features into the system by identifying the distribution (or combination) of functional components of safety features in the design space.

B. Survey of Safety Features in Medical Robotics

As part of the validation of SDV, we performed a literature survey on the safety designs in the medical robotics domain. The objective was to collect and identify a set of safety features that have been frequently used in the domain or have been reported to be effective based on actual experiences. We reviewed about 90 *academic* articles starting from 1985 [18] up to 2013 [19], [6], focusing on safety, safety designs, and safety methods. Through our literature review, we were able to identify a set of safety features that have been repeatedly used across a variety of medical robot systems in history, as in Table I. Although we found a large set of safety features, only a few are presented here due to page limitations.

TABLE I: Safety features most frequently used in the medical robotics domain

Safety Feature	Examples
Visualization (e.g., display of plans, current progress)	[20], [21]
Force threshold checks	[22], [23]
Low speed or motor torque (i.e., slow motion)	[24], [25]
E-pause and/or E-stop	[11], [26]
Dynamic constraints (e.g., safety volume, virtual fixture)	[27], [28]
Redundant sensors (e.g., dual encoders)	[29], [30]

C. Examples: Safety Features of ROBODOC

To illustrate how to apply SDV to an actual system, we selected a commercial robot system for orthopaedic surgery, the ROBODOC® system [31], [32], as a case study. This system has a solid set of safety features that obtained FDA approval and CE marking, and has been in clinical use since 1992. Most importantly, a relatively large number of academic publications about the system and safety designs are available.

The safety features in Table I are also found in the ROBODOC system and we applied SDV to them, as in Fig. 4. ROBODOC fits the canonical system model where the low-level control is performed on dedicated joint control boards, the high-level control (e.g., Cartesian motion and force control) is a real-time loop on a PC, and the application (workflow) runs in non-real-time on the same PC. The surgeon interfaces with the system via graphical menus and a hand-held control pendant; in addition to the buttons for selecting menu items, the pendant includes pause and stop buttons that freeze robot motion and turn off robot motor power, respectively. In order to complete the SDV, it was first necessary to define some conventions. For example, many safety features rely on hardware to measure a physical quantity, such as position or force, and this feedback is acquired by one of the software layers (typically low-level or high-level control). In this case, we place solid dots in the monitoring (M) column corresponding to the rows for the hardware layer (HW) and the appropriate software layer (e.g., LC or HC).

The force limit checking (Fig. 4a) shows a typical run-time safety mechanism where monitoring is done at the hardware layer, whereas detection and reaction occur at the control layer, and recovery relies on the human. Compared to this, the redundant sensor (e.g., encoder mismatch) has more mechanisms implemented in the low-level control layer (Fig. 4b). ROBODOC included both an E-stop and an E-pause, where the former is initiated by hardware and the latter by software. On SDV, it is straightforward to discern the differences between them, as in Figs. 4c and 4d. The motor speed limit is implemented as part of the high-level control loop (Fig. 4e) because that is where the trajectory generation was performed (this safety feature limits the commanded joint speed, which could otherwise become excessive near a kinematic singularity); for torque-controlled robots, a torque limit would be used instead. The safety volume (Fig. 4f) appears similar to the motor speed or torque limits, but also relies on the monitoring mechanism in the hardware layer to check the actual robot position measured by the encoders.

In Fig. 4, we found a "pattern" in the reaction and recovery mechanisms of the safety features. This was, of course, part of the ROBODOC system design and SDV "captured" it. For example, we note that all reactions involve at least one control layer (typically to stop motion or initiate power-off), but also affect the workflow by stopping the normal sequence to display an error message to the user. In all cases shown, the human (surgeon) is involved in the recovery action. For the force threshold check, the workflow also initiated part of the recovery by automatically backing away along the measured force direction. It is possible for recovery to be performed without human involvement, where the system silently recovers from an unsafe condition, but that was not the case for any of the above safety features.

	M	D	R	Re
HU				●
WF			●	●
HC	●	●	●	
LC		○		
HW	●			

(a) Force limit check

	M	D	R	Re
HU				●
WF			●	
HC	○	●	●	
LC	○	●	●	
HW	●			

(b) Redundant sensors

	M	D	R	Re
HU				●
WF			●	
HC			●	
LC			●	
HW	●	●	●	

(c) E-Stop (hardware-based)

	M	D	R	Re
HU				●
WF			●	
HC	●	●	●	
LC	○	○		
HW	●			

(d) E-Pause (software-based)

	M	D	R	Re
HU				●
WF			●	
HC	●	●	●	
LC	○	○	○	
HW				

(e) Motor speed or torque limits

	M	D	R	Re
HU				●
WF			●	
HC	●	●	●	
LC				
HW	●			

(f) Safety volume

Fig. 4: Representation of safety features of ROBODOC using Safety Design View (●: primary choice, ○: possible alternatives). The horizontal and vertical axes correspond to the mechanism view and the system view, respectively.

V. Discussion

Our approaches to understanding and representing safety can be summarized as three elements: the mechanism view, the system view, and SDV. They are designed to better present safety features of medical robot systems by simultaneously identifying run-time safety mechanisms and capturing design decisions on deployment options. This section discusses some of the design details of these elements as well as some limitations that we experienced throughout this work.

We note that not all safety features can be represented by the four components of the mechanism view. For example, one possible safety feature is to design the robot with low-power motors so that it has limited speed and/or torque. This is essentially a safety feature that is implemented by changing a *property* of the system (it is different from a software-imposed speed or torque limit, as presented in Fig. 4e). In our experience, these types of safety features are typically confined to hardware design properties and are therefore outside the scope of this work.

The current definition of the system view does not yet consider another type of human–*patients*–because the focus is on engineered systems. However, patient safety is also a crucial aspect and there is a body of work in this area, i.e., physical interactions between the robot and human [33], [34]. One possible extension may be to add another layer (such as "Patient") to the system view, below the hardware layer.

The locations of black dots in the design-space of safety features (SDV) captures design decisions about the deployment options of safety features, which reflects system designers' experience and expertise. One possible use of SDV is to document and collect representative safety features using SDV, and establish "canonical templates" of such safety features, which describe or define how to design, implement, and deploy safety features as "best practices" or guidelines.

In the design-space of safety features, one limitation is that it can capture the data and control flow *horizontally*, but is hard to do *vertically*. The mechanism view has a notion of "flow" from monitoring to recovery, but the system view does not. Sometimes safety features behave in specific orders and may need specific timing requirements. For example, if the workflow layer detects a safety violation, it may initiate part of the reaction, but would likely need to request the control layer to stop motion or power off the motors. The ability to capture this sequence of actions between the system layers is a possible extension to SDV.

Through this work, we noticed that the effectiveness of

SDV depends on the SDV user's experience and the degree of understanding of safety features. Because SDV is based on the concept of abstraction (of mechanisms) and layers (of deployment options), the deeper the understanding is, the more effective SDV becomes.

Despite its limitation in terms of expressiveness, SDV was effective and helpful for clearly describing, documenting, and conveying the idea and design of safety features, based on our experience. Especially, representing safety features of ROBODOC using SDV (Sec. IV-C) led to in-depth design discussions of its current safety design.

VI. Conclusion and Future Works

We presented the Safety Design View (SDV), a conceptual framework that can capture and describe both the design-time and the run-time characteristics of safety features of medical robot systems in a systematic and structured manner. SDV is based on two views, the mechanism view and the system view, each dealing with safety mechanisms and design decisions on the deployment of safety features. The goal of SDV is to: (1) explicitly and intuitively describe safety features in a consistent and structured manner, (2) collect "good" practices on the design of safety features, and (3) facilitate sharing of knowledge and experience on safety within the community.

There are active discussions for safety standards of medical robotics in the community. Prior to these discussions, or for more effective discussions, it is our hope that such a common ground as SDV could be helpful. Also, it is possible that SDV may become an additional method for conveying the safety design within a medical device company, and between the company and regulatory agencies.

We are currently working in two directions: one is to further elaborate and improve the design and definition of SDV, and the other is to review and organize our safety survey results into a set of "canonical templates" of safety features.

References

[1] N. G. Leveson, *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[2] ——, "Safety as a system property," *ACM Communications*, vol. 38, no. 11, p. 146, Nov. 1995.

[3] R. Taylor, H. Paul, P. Kazanzides, B. Mittelstadt, W. Hanson, J. Zuhars, B. Williamson, B. Musits, E. Glassman, and W. Bargar, "Taming the bull: Safety in a precise surgical robot," in *Intl. Conf. on Advanced Robotics (ICAR)*, vol. 1, Jun. 1991, pp. 865–870.

[4] B. L. Davies, *A discussion of safety issues for medical robots*, ser. Computer-Integrated Surgery. MIT Press, 1996, pp. 287–298.

[5] M. D. O'Toole, K. Bouazza-Marouf, D. Kerr, M. Gooroochurn, and M. Vloeberghs, "A methodology for design and appraisal of surgical robotic systems," *Robotica*, vol. 28, no. 02, pp. 297–310, 2010.

[6] P. Dombre, E. Poignet and F. Pierrot, *Design of Medical Robots*. John Wiley & Sons, Inc., 2013, ch. 5, pp. 141–176.

[7] M. Y. Jung and P. Kazanzides, "Run-time Safety Framework for Component-based Medical Robots," in *Medical Cyber Physical Systems Workshop, CPSWeek*, 2013.

[8] W. Ng and C. Tan, "On safety enhancements for medical robots," *Reliability Eng. & Sys. Safety*, vol. 54, no. 1, pp. 35–45, 1996.

[9] B. Fei, W. S. Ng, S. Chauhan, and C. K. Kwoh, "The safety issues of medical robotics," *Reliability Engineering & System Safety*, vol. 73, no. 2, pp. 183–192, 2001.

[10] A. Kapoor, A. Deguet, and P. Kazanzides, "Software components and frameworks for medical robot control," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2006, pp. 3813–3818.

[11] P. Kazanzides, J. Zuhars, B. Mittelstadt, B. Williamson, P. Cain, F. Smith, L. Rose, and B. Musits, "Architecture of a surgical robot," in *IEEE Intl. Conf. on Systems, Man and Cybernetics*, vol. 2, Oct. 1992, pp. 1624–1629.

[12] G. Guthart and J. Salisbury, J.K., "The Intuitive$^{\text{TM}}$ Telesurgery System: Overview and Application," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 1, 2000, pp. 618–621.

[13] M. Visinsky, I. Walker, and J. Cavallaro, "Layered dynamic fault detection and tolerance for robots," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 1993, pp. 180–187.

[14] D. Kortenkamp and R. Simmons, *Robotic Systems Architectures and Programming*. Springer, 2009, pp. 187–206.

[15] P. Kruchten, "The 4+1 View Model of Architecture," *IEEE Software*, vol. 12, pp. 42–50, 1995.

[16] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods–an introduction," *Control Eng. Practice*, vol. 5, pp. 639–652, 1997.

[17] P. Kazanzides, "Safety design for medical robots," in *IEEE Intl. Conf. on Eng. in Medicine and Biology Society*, Sep. 2009, pp. 7208–7211.

[18] H. M. Shao, J. Y. Chen, T. K. Truong, I. S. Reed, and Y. S. Kwoh, "A New CT-Aided Robotic Stereotaxis System," in *Annu. Symp. Comput. Appl. Med Care.*, vol. 13, Nov. 1985, pp. 668–672.

[19] L. Sanchez, M. Le, K. Rabenorosoa, C. Liu, N. Zemiti, P. Poignet, E. Dombre, A. Menciassi, and P. Dario, "A Case Study of Safety in the Design of Surgical Robots: The ARAKNES Platform," in *Intelligent Autonomous Systems 12*, ser. Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, 2013, vol. 194, pp. 121–130.

[20] J. Troccaz and Y. Delnondedieu, "Semi-active guiding systems in surgery. A two-DOF prototype of the passive arm with dynamic constraints (PADyC)," *Mechatronics*, vol. 6, no. 4, pp. 399–421, 1996.

[21] P. Bast, A. Popovic, T. Wu, S. Heger, M. Engelhardt, W. Lauer, K. Radermacher, and K. Schmieder, "Robot- and computer-assisted craniotomy: resection planning, implant modelling and robot safety," *Intl. J. of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 2, pp. 168–178, 2006.

[22] P. Kazanzides, J. Zuhars, B. Mittelstadt, and R. Taylor, "Force sensing and control for a surgical robot," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 1, May 1992, pp. 612–617.

[23] E. Dombre, G. Duchemin, P. Poignet, and F. Pierrot, "Dermarob: A safe robot for reconstructive surgery," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 5, pp. 876–884, Oct. 2003.

[24] M. Jakopec, F. Rodriguez y Baena, S. Harris, P. Gomes, J. Cobb, and B. Davies, "The hands-on orthopaedic robot "Acrobot": Early clinical trials of total knee replacement surgery," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 5, pp. 902–911, Oct. 2003.

[25] U. Laible, T. Bürger, and G. Pritschow, "A fail-safe dual channel robot control for surgery applications," *Safety Science*, vol. 42, no. 5, pp. 423–436, 2004.

[26] J. Guiochet and A. Vilchis, "Safety analysis of a medical robot for tele-echography," *Proc. of the 2nd IARP/IEEE RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments*, pp. 217–227, 2002.

[27] S. Ho, R. Hibberd, and B. Davies, "Robot assisted knee surgery," *IEEE Eng. in Med. & Bio. Mag.*, vol. 14, no. 3, pp. 292–300, May/Jun. 1995.

[28] W.-H. Zhu, S. Salcudean, S. Bachmann, and P. Abolmaesumi, "Motion/force/image control of a diagnostic ultrasound robot," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 1580–1585.

[29] E. Degoulange, L. Urbain, P. Caron, S. Boudet, J. Gariepy, J.-L. Megnien, F. Pierrot, and E. Dombre, "HIPPOCRATE: an intrinsically safe robot for medical applications," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, Oct. 1998, pp. 959–964.

[30] D. Engel, J. Raczkowsky, and H. Worn, "A safe robot system for craniofacial surgery," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2001, pp. 2020–2024.

[31] P. Kazanzides, P. W. Cain, and H. A. Wasti, "Distributed architecture for a fail-safe robot system," in *Proc. of the Signal Processing Applications Conference & Exhibition (DSPx)*, San Jose, CA, Mar. 1996.

[32] P. Kazanzides, "Robot Assisted Surgery: The ROBODOC® Experience," in *Intl. Symp. on Robotics*, vol. 30, Tokyo, Japan, 1999, pp. 281–286.

[33] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *Intl. Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1507–1527, 2009.

[34] S. Haddadin, S. Haddadin, A. Khoury, T. Rokahr, S. Parusel, R. Burgkart, A. Bicchi, and A. Albu-Schaffer, "A truly safely moving robot has to know what injury it may cause," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5406–5413.