# Run-time Monitoring for Robot-Assisted Surgery

Kristina Gogoladze[1], Natasha Alechina[4,1], Zhaoyang Jacopo Hu[2], Haozheng Xu[2], Romy van Jaarsveld[3] and Jelle P. Ruurda[3]

[1] *Utrecht University, Utrecht*
[2] *Imperial College London, London*
[3] *University Medical Centre Utrecht, Utrecht*
[4] *Open Universiteit, Heerlen*

**Abstract**
We present a proposal to support surgeons during robot-assisted surgery. Our approach is based on run-time monitoring for properties that should hold during the surgery, and issuing a warning to the surgeon if they are violated. We outline the general approach involving properties that require both visual input from the cameras as well as kinematic information obtained from the robot's manipulators. Then we describe preliminary work on monitoring using purely kinematic information.

**Keywords**
Runtime verification, Robot-assisted surgery, Linear temporal logic

## 1. Introduction

Robot-assisted surgery involves the surgeon remotely operating a surgical robot that performs operations on the patient. The surgeon has a magnified view of the area of surgery. The robot can have multiple arms and use various tools with great precision. Robot-assisted surgery is increasingly being adopted as the surgical procedure of choice for a wide range of operations [1]. Robot-assisted surgery has several advantages over traditional, open, surgery, including improved dexterity, tremor filtering, three-dimensional magnified vision, and smaller incisions. Due to smaller incisions, it leads to faster recovery and better patient outcomes. However, robot-assisted surgery is complex and requires considerable training to become proficient. For example, Robot-Assisted Minimally Invasive Esophagectomy (RAMIE) requires over 100 training cases [1]. Even trained surgeons may still make mistakes or come close to making a mistake, such as touching or damaging anatomical structures not involved in the operation. The reasons for this include obscured vision, disorientation due to the highly magnified view, misrecognition of anatomy, lack of tactile feedback, and fatigue. The challenge of improving surgical performance is very important [2].

This paper is the first step in a project to improve patient outcomes by supporting surgeons during robot-assisted surgery. We aim to develop a run-time verification system that warns a surgeon if the robot manipulators are getting too close to, e.g., the aorta, or that the pressure exerted is too high, or the movement too fast or uneven. In order to do this, we need to be able to specify properties of interest (such as, 'the tool should not touch the aorta') in a precise formal language.

Run-time verification has been successfully used in a wide range of applications, including software systems, avionics, and autonomous space robotics [3, 4, 5]. There is also work on surgery monitoring using Siamese neural networks and transformers [6, 7]. However, run-time verification of robot-assisted surgery that involves precise formal guarantees and can be used for any monitorable property has not previously been attempted, although there is work on verification applied to robotic surgery, notably [8, 9].

*OVERLAY 2024, 6th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, November 28–29, 2024, Bolzano, Italy*
k.gogoladze@uu.nl (K. Gogoladze); n.a.alechina@uu.nl (N. Alechina); jh920@ic.ac.uk (Z. J. Hu); hx819@ic.ac.uk (H. Xu); R.C.vanJaarsveld-3@umcutrecht.nl (R. van Jaarsveld); J.P.Ruurda@umcutrecht.nl (J. P. Ruurda)

Run-time monitoring of robot-assisted surgery involves significant challenges, including: the complexity of the behaviour (of the robot manipulated by a human); the complexity of the environment (soft tissues in a human body, which move with every breath and heartbeat, and change during surgical dissection); and the need to come up with a specification language that is sufficiently expressive to allow the statement of medically relevant properties while at the same time admitting efficient verification. We focus on the case study of Robot-Assisted Minimally Invasive Esophagectomy (RAMIE) [1], but the approach will be applicable to other surgical procedures.

## 2. Properties of interest

One of the objectives of our research is to identify and describe relevant and specific properties that need to be monitored for. We describe several types of properties within the RAMIE procedure that are important to be monitored in order to obtain a correct execution of the procedure while minimizing postoperative complications or injuries. The current list of potential properties of interest has been composed in consultation with the medical experts participating in the project. In future work, we plan to evaluate and possibly expand the list through a survey of surgeons training to perform the RAMIE procedure.

The first kind of properties to be checked is avoiding critical anatomical structures where possible. During the procedure, a lot of critical anatomical structures are exposed from their surrounding tissue that also serve as a protective layer. During the procedure lymph nodes are removed around the anatomical structures to extract cancer cells as well as to get a clear visual of the diseased anatomical structure parts that need to be removed. Critical anatomical structures like the aorta, azygos vein, trachea, and laryngeal recurrent nerve are easily damaged which can lead to serious (post operative) complications and injuries, some even leading to mortality. Therefore, it is necessary to keep away from critical anatomical structures when possible during a particular phase in the procedure.

Another kind of properties to be monitored is the motions and movements the surgeon makes during the procedure. Hasty or hesitant movements and motions can lead to scraping or touching critical anatomical structures. When a surgeon exhibits this kind of behaviour, alerting them could lead to them reflecting on their execution and slow down and be more precise if necessary. We could monitor for too slow or too fast movements, and in general for patterns of movement: either try to compare patterns for different people, or learn what the movement pattern is for a given surgeon and subsequently compare the current surgery's movements with already established movements for a particular surgeon in order, for example, to determine if the surgeon is suddenly tired or tense.

During the surgery, the surgeon should not stop for more than a few seconds. If the surgeon stops, it may be a sign that something is going wrong. Therefore, 'stopping for more than X seconds' is also a property to be checked.

A very important property associated with the tools in the hands of the robot is whether they are in the camera view. This is important because if we do not see the instruments, then the patient's safety cannot be guaranteed. It is important to clearly and fully see the working field (area the tools are working in) during surgery. Sometimes it can be forgotten to focus the camera view centrally on the working field. When working in the outer rim, a smaller area can be seen around the entire working field. This could lead to damaging surrounding structures. Therefore, properties such as 'working field in outer rim of video view' and 'instrument completely out of view' are very important to monitor for.

When performing robot-assisted surgeries, the operating surgeon does not feel resistance if the instrument touches anything. If the instruments touch each other, there is a possibility that they are actually pressing against each other with great force, in which case they can slip and cause irreparable harm to the patient. This is why it is important to always check that instruments are not touching each other.

Another possible property to check for is whether suturing is being done in the correct direction; namely, at one end of the anastomosis (e.g., the esophagus) the needle goes from outermost to innermost layer; on the other end (e.g., gastric tube) the needle goes from the innermost to the outermost layer.

Finally, correctly completing surgical phases before going on to the next one is crucial for the correct execution of the RAMIE procedure. Monitoring the full completion of surgical phases can be helpful to surgeons as a memory aid during the procedure.

## 3. Expressing Properties in Linear Temporal Logic

In run-time verification, properties to monitor are often expressed in Linear Temporal Logic (LTL) [10]. This formal language is intended for describing constraints on runs of the system (sequences of states of the system). It can say that some statement $\varphi$ holds in the Next state, that $\varphi$ holds Globally (in every state) and that some statement $\psi$ holds until $\varphi$ becomes true ($\psi$ Until $\varphi$). A system execution is formally a simple *linear trace* — a sequence of events that capture a behaviour of the monitored system we are interested in analysing.

We consider a variant of LTL interpreted over finite traces [11, 12], called LTL$_f$. LTL$_f$ is formally defined as follows. Let $\mathcal{AP}$ be a set of atomic propositions, then the syntax of a formula in LTL$_f$ is

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \mathcal{X}\phi \mid \phi\mathcal{U}\psi$$

where $p \in \mathcal{AP}$. We use the standard abbreviations such as $\phi \vee \psi ::= \neg(\neg\phi \wedge \neg\psi)$ and $\phi \rightarrow \psi ::= \neg\phi \vee \psi$, as well as $\Diamond\phi ::= \top\mathcal{U}\phi$ (eventually) and $\Box\phi ::= \neg\Diamond\neg\phi$ (globally).

A *state* is a set of atomic propositions from $\mathcal{AP}$ (intuitively, a state is a set of atomic propositions that are true there). A *finite trace* $\pi$ is a finite sequence of states. We denote by $|\pi|$ the length of $\pi$, by $\pi[i]$ the $i$th state on $\pi$ (counting from 0) and by $\pi_i$ the trace beginning at position $i$ of $\pi$.

The notion of LTL$_f$ formula $\phi$ being $true$ at a trace starting at position $i$, denoted as $\pi_i \models \phi$, is defined inductively as follows:

$\pi_i \models p$ iff $p \in \pi[i]$;

$\pi_i \models \neg\phi$ iff $\pi_i \not\models \phi$;

$\pi_i \models \phi \wedge \psi$ iff $\pi_i \models \phi$ and $\pi_i \models \psi$;

$\pi_i \models \mathcal{X}\phi$ iff $i < |\pi|$ and $\pi_{i+1} \models \phi$;

$\pi_i \models \phi\mathcal{U}\psi$ iff for some $j$ such that $i \leq j \leq |\pi|$, we have that $\pi_j \models \psi$ and for every $k$, $i \leq k < j$ it holds that $\pi_k \models \phi$.

Below we give some examples of how some properties introduced in the previous section can be expressed in LTL$_f$.

An example of a 'safe distance to the anatomy' property is 'the distance to aorta should always be at least $d$'. This can be expressed in LTL$_f$ as $\Box(distance > d)$. Note that $distance > d$ is an atomic proposition. The ability to monitor this property relies crucially on being able to identify anatomical structures such as the aorta during surgery, and there have been considerable advances in research in this field, see for example [13, 14].

A similar property is monitoring for too high speed of the tool tip: $\Box(speedToolTip < s)$.

A related problem is how much 'slack' to leave in the safe interval of the variables' values. To come back to the example of the distance to aorta, signalling the violation when the distance is 0 is correct but not helpful. Setting the set distance to a larger value, e.g. 1 cm, is also not helpful, because it may be perfectly safe and necessary to position the tool tip at a shorter distance to the aorta. In general a good formulation would require taking into the consideration the phase of the surgery, the speed and direction of movement of the tool tip as well as the distance.

An additional challenge is in formally defining properties which are to do with human behaviour rather than the values of physical characteristics of the robotic tools (speed, position, etc.). For example, properties such as the surgeon's movements being too hesitant, or movements indicating that the surgeon is tired. These types of warnings are considered by surgeons to be very useful, and their timing is less critical than the ‘distance to aorta’ type of properties. However, as far as we know, there is no work on formal run-time monitoring for this kind of property. For example, one way of specifying formally the property that ‘the surgeon’s movement is hesitant’, could be stating that the speed of the tool tip should not be above $s$, below $s$, and then above it again:

$$\Box\neg((speedToolTip > s) \wedge \mathcal{X}(speedToolTip < s) \wedge \mathcal{X}\mathcal{X}(speedToolTip > s))$$

Similarly properties suggesting that the surgeon is slowing down or is making more aggressive movements can be specified. The specification of such properties may be specific to a particular surgeon. It is well known that each surgeon is different, and what is too aggressive for one surgeon may be the usual manner of operating for another very experienced and highly successful surgeon. This problem is related to the problem of how to time a warning so that a violation of the property can be prevented, however in this case we must decide when the movement has become too slow or too aggressive for this particular surgeon.

## 4. Experiments with kinematic information

The most successful device used for RAMIE is the da Vinci Surgical System (Intuitive Surgical Inc., Sunnyvale, CA). In the mid-2010s, the first generation of the da Vinci robot was taken out of service, since it was impractical to be serviced and supplied anymore. However, these robotic platforms were still functional, and could be used for research. The development of the da Vinci Research Kit (dVRK) was started at the Johns Hopkins University, and in a few years, an active community has gathered with more than 30 setups worldwide. The dVRK consists of open-source, custom-built hardware controllers and software elements to make possible the programming of the attached da Vinci arms.

Considering that robot manufacturing companies do not provide access to robot sensors, our main methods of obtaining data from tools will be visual and kinematic information. Because we do not have reliable real-time image segmentation yet, in this paper we will focus on kinematic information from the Patient Side Manipulators (PSMs) of a dVRK and the properties that can be expressed using only such information. Using kinematic data we can monitor all properties that do not involve identifying anatomical structures from visual information. This includes, for example, the following properties: hesitant movements, too slow or too fast movements, patterns of movements, conflicts of instruments, instruments in the camera view, and suturing in the wrong direction.

Let us consider an experiment in runtime monitoring during suturing. The data we get from the sensors contains a time stamp, positions of the left and right PSMs of the dVRK and of the sigma.7 hand interfaces which we adopted to control the dVRK, rotation matrices of the PSMs, orientations of the sigma.7s, translational and rotational velocities of the PSMs and the sigma.7s, and gripper angles of the PSMs and the sigma.7s.

Since the cartesian coordinates of each PSM are given in respect to their own base, we place their origins in a global coordinate knowing that their location and distance should be approximately represented when their end-effectors are the closest during the hand-off of the needle.

The monitoring was performed of 2 robotic suturing throws (1 throw = 1 stitch). The setup was similar to [15], which comprised of the dVRK PSMs that integrate two sigma.7 hand interfaces (Force Dimension, Nyon, Switzerland) to allow the user to control the robotic platform.

We augment the raw data that we get from the sensors: in every state, in addition to storing $x$, $y$ and $z$ coordinates of the tool tip, we also store the coordinate values we had in the previous state (except the very first state, we can set it to be the same there), we denote them by $x^{-}$, $y^{-}$ and $z^{-}$. We also store the velocity for the current state, as well as other parameters we might need for expressing the properties we are monitoring. In addition to adding the necessary parameters to the states, we remove all unused variables received by the sensors. Thus, we get the sequence of states we need.

Having this augmented data, the setup is as follows. The sequence of states is generated as above, with a state for the data received at each time stamp. If we want to monitor, for example, for the correct direction of suturing, knowing all the coordinates, positions, and speed in the current time point and in the previous ones, we can tell whether the direction is correct. Let us look at an easy example of simple continuous suturing. The direction of the stitches here is along the wound. Let $X$ be an axis parallel to the wound (if wound is a straight incision), so the correct direction of suturing is along $X$. We also have $x$ coordinates of the tool tip during suturing. Specifying the property (the correct direction of suturing), we want to make sure that the difference between the subsequent moments (coordinates at subsequent moments) is positive. Assume we are performing suturing and we have a way of determining when knot tying happens, then we can express the property in LTL$_f$ as follows:

$$(ToolTip_x > ToolTip_{x^-})\mathcal{U}(KnotTied),$$

where $ToolTip_x$ is the $x$ coordinate of the tool tip and $ToolTip_{x^-}$ is the $x$ coordinate of the tool tip at the previous state. The property is described in LTL$_f$, then we can use one of the existing software tools that will monitor this property in real time, for example the one discussed in [16, 17][1].

To check that the surgeon is not stopping for more than $m$ steps, we have to verify that at least one coordinate is changing every $m$ steps:

$$\Box\neg(ToolTip_x = ToolTip_{x^-} \wedge \mathcal{X}(ToolTip_x = ToolTip_{x^-} \wedge \ldots \wedge \mathcal{X}(ToolTip_x = ToolTip_{x^-} \underbrace{)\ldots)}_{m-1} \wedge$$
$$\wedge ToolTip_y = ToolTip_{y^-} \wedge \mathcal{X}(ToolTip_y = ToolTip_{y^-} \wedge \ldots \wedge \mathcal{X}(ToolTip_y = ToolTip_{y^-} \underbrace{)\ldots)}_{m-1} \wedge$$
$$\wedge ToolTip_z = ToolTip_{z^-} \wedge \mathcal{X}(ToolTip_z = ToolTip_{z^-} \wedge \ldots \wedge \mathcal{X}(ToolTip_z = ToolTip_{z^-} \underbrace{)\ldots)}_{m-1}).$$

We can also verify that the tools are not outside of the camera view. If for simplicity we assume that the $Z$ axis is perpendicular to the camera view, then we only have to worry about the other coordinates. Assume the bottom left corner has coordinates $(0, 0)$ and the top right corner – $(A, B)$, then the property of interest looks like this

$$\Box(ToolTip_x > 0 \wedge ToolTip_x < A \wedge ToolTip_y > 0 \wedge ToolTip_y < B).$$

When the $Z$ axis is not orthogonal to the camera view, we have to calculate the projections for the given camera angle.

## 5. Future Work

Addressing the problems of setting the safe interval for the variable values, and of different styles of different surgeons is one of the challenges facing our project. We plan to address this challenge by learning the correct parameters (or the correct automaton corresponding to the property) by using techniques for learning automata (deterministic finite automata in this case) from data (in our case, video records of surgeries), see, for example, [18]. We expect that these techniques will enable us to learn, e.g., an individual surgeon's 'normal' movement and speed characteristics. Once we have the automaton representing the surgeon's normal movements, we will attempt to monitor for unusual movement patterns.

We intend to experiment with different formalisms to use in stating properties to be verified and for synthesising monitors from them. It is accepted in the run-time verification community that expressing constraints on history (past) rather than future-directed constraints is often more intuitive and more efficient to monitor for [19, 20]. We will investigate the past version of LTL$_f$ and study its computational properties and suitability for expressing properties related to robot-assisted surgery.

Evaluation of the prototype monitoring system will be carried out by showing the output of the system to observer surgeons who are not carrying out the surgery but are seeing the same view from the camera as the operating surgeon, and asking them to rate the usefulness of the warnings.

**Acknowledgements** We gratefully acknowledge funding by NWO project *Run-time verification for robot-assisted surgery*, OCENW.M.21.377.

[1] Note that this property is not monitorable in LTL, but is monitorable in LTL$_f$.

## References


[1] J. P. Ruurda, P. C. van der Sluis, S. van der Horst, R. van Hilllegersberg, Robot-assisted minimally invasive esophagectomy for esophageal cancer: A systematic review, J Surg Oncol. 112 (2015) 257–265. doi:doi:10.1002/jso.23922.

[2] J. W. Suliburk, Q. M. Buck, C. J. Pirko, N. N. Massarweh, N. R. Barshes, H. Singh, T. K. Rosengart, Analysis of human performance deficiencies associated with surgical adverse events, JAMA Network Open 2 (2019). doi:doi:10.1001/jamanetworkopen.2019.8067.

[3] E. Bartocci, Y. Falcone (Eds.), Lectures on Runtime Verification - Introductory and Advanced Topics, volume 10457 of *Lecture Notes in Computer Science*, Springer, 2018. URL: https://doi.org/10.1007/978-3-319-75632-5. doi:10.1007/978-3-319-75632-5.

[4] A. Ferrando, R. C. Cardoso, M. Fisher, D. Ancona, L. Franceschini, V. Mascardi, ROSMonitoring: A runtime verification framework for ROS, in: A. Mohammad, X. Dong, M. Russo (Eds.), Towards Autonomous Robotic Systems - 21st Annual Conference, TAROS 2020, volume 12228 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 387–399.

[5] B. Hertz, Z. Luppen, K. Y. Rozier, Integrating runtime verification into a sounding rocket control system, in: A. Dutle, M. M. Moscato, L. Titolo, C. A. Muñoz, I. Perez (Eds.), NASA Formal Methods - 13th International Symposium, NFM 2021, Virtual Event, May 24-28, 2021, Proceedings, volume 12673 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 151–159. doi:10.1007/978-3-030-76384-8.

[6] Z. Shao, J. Xu, D. Stoyanov, E. B. Mazomenos, Y. Jin, Think step by step: Chain-of-gesture prompting for error detection in robotic surgical videos, 2024. URL: https://arxiv.org/abs/2406.19217. arXiv:2406.19217.

[7] Z. Li, K. Hutchinson, H. Alemzadeh, Runtime detection of executional errors in robot-assisted surgery, in: 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 3850–3856. doi:10.1109/ICRA46639.2022.9812034.

[8] D. Bresolin, L. Geretti, R. Muradore, P. Fiorini, T. Villa, Formal verification of robotic surgery tasks by reachability analysis, Microprocess. Microsystems 39 (2015) 836–842. URL: https://doi.org/10.1016/j.micpro.2015.10.006. doi:10.1016/j.micpro.2015.10.006.

[9] D. Bresolin, L. Geretti, R. Muradore, P. Fiorini, T. Villa, Formal verification applied to robotic surgery, Lecture Notes Control Inf Sci. 456 (2015) 347–355. doi:doi:10.1007/978-3-319-10407-2_40.

[10] A. Pnueli, The temporal logic of programs, in: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (SFCS-77), IEEE, 1977, pp. 46–57.

[11] G. D. Giacomo, M. Y. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: F. Rossi (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013, IJCAI/AAAI, 2013, pp. 854–860.

[12] S. Cerrito, M. C. Mayer, S. Praud, First order linear temporal logic over finite time structures, in: H. Ganzinger, D. McAllester, A. Voronkov (Eds.), Logic for Programming and Automated Reasoning, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 62–76.

[13] A. S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on MRI, Zeitschrift für Medizinische Physik 29 (2019).

[14] R. den Boer, C. de Jongh, W. Huijbers, T. Jaspers, J. Pluim, R. van Hillegersberg, M. van Eijnatten, J. Ruurda, Computer-aided anatomy recognition in intrathoracic and -abdominal surgery: a systematic review, Surgical Endoscopy 36 (2022) 8737–8752. doi:10.1007/s00464-022-09421-5.

[15] A. Saracino, A. Deguet, F. Staderini, M. N. Boushaki, F. Cianchi, A. Menciassi, E. Sinibaldi, Haptic feedback in the da Vinci Research Kit (dvrk): A user study based on grasping, palpation, and incision tasks, The International Journal of Medical Robotics and Computer Assisted Surgery 15 (2019) e1999.

[16] L. Geatti, A. Montanari, N. Saccomanno, Towards machine learning enhanced LTL monitoring, in: Proceedings of the 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis hosted by the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023), 2023, pp. 13–19.

[17] T. Yamaguchi, B. Hoxha, D. Ničković, RTAMT – Runtime Robustness Monitors with Application to CPS and Robotics, International Journal on Software Tools for Technology Transfer 26 (2024) 79–99. doi:10.1007/s10009-023-00720-3.

[18] E. Abadi, R. I. Brafman, Learning and solving regular decision processes, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 1948–1954. URL: https://doi.org/10.24963/ijcai.2020/270. doi:10.24963/ijcai.2020/270, main track.

[19] O. Lichtenstein, A. Pnueli, L. D. Zuck, The glory of the past, in: R. Parikh (Ed.), Logics of Programs, volume 193 of *Lecture Notes in Computer Science*, Springer, 1985, pp. 196–218.

[20] L. Geatti, N. Gigante, A. Montanari, G. Venturato, Past Matters: Supporting LTL+Past in the BLACK Satisfiability Checker, in: C. Combi, J. Eder, M. Reynolds (Eds.), 28th International Symposium on Temporal Representation and Reasoning, TIME 2021, September 27-29, 2021, Klagenfurt, Austria, volume 206 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 8:1–8:17. URL: https://www.dagstuhl.de/dagpub/978-3-95977-206-8.