Monitoring robot-assisted surgery using kinematics

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

[1] Utrecht University, The Netherlands
[2] Open Universiteit, The Netherlands
[3] Imperial College London, United Kingdom
[4] University Medical Center Utrecht, The Netherlands

We present an approach to improving safety of surgical procedures based on automated run-time monitoring [1] for properties that should hold during the surgery. In this abstract, we describe monitoring using kinematic information obtained from the robot's manipulators (rather than visual input from the robot cameras).

Run-time verification has been successfully used in a wide range of applications, including software systems, avionics, and autonomous space robotics [1, 5, 8]. There is also work on surgery monitoring using Siamese neural networks and transformers [12, 9]. 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 [3, 2].

We compiled a list of properties within the RAMIE [10] procedure that are important to be monitored for in order to obtain a correct execution of the procedure while minimizing postoperative complications or injuries.

Some of the examples of the properties that need to be monitored for include: the surgeon's movements should not exceed certain speed, the surgeon should not stop for more than a few seconds, the tools in the hands of the robot should be in the camera view, suturing should be done in the correct direction.

In run-time verification, properties to monitor are often expressed in Linear Temporal Logic over finite traces (LTL_f) [7, 4]. 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 φ holds in the Next state ($\mathcal{X}\phi$), that φ holds Globally ($\Box\phi$) and that some statement ψ holds until φ becomes true ($\psi\mathcal{U}\varphi$). 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.

As an example, monitoring for too high speed of the tool tip can be expressed as: $\Box(\mathit{speedToolTip} < s)$. Note that $\Box(\mathit{speedToolTip} < s)$ is a proposition.

The most successful device used for RAMIE is the da Vinci Surgical System (Intuitive Surgical Inc., Sunnyvale, CA). Open-source, custom-built hardware controllers and software elements (dVRK) based on the first generation of the da Vinci robot are now used for research. We briefly consider an experiment in runtime monitoring during suturing using dVRK. 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 that 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.

The monitoring was performed of 2 robotic suturing throws (1 throw = 1 stitch). The setup was similar to [11], 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, 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 disregard other input variables. This gives us a sequence of states on which to evaluate the formulas during monitoring. 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 have a way of determining when knot tying happens, then we can express the property as follows:

$$(\mathit{ToolTip}_x > \mathit{ToolTip}_{x^-})\mathcal{U}(\mathit{KnotTied}),$$

where $\mathit{ToolTip}_x$ is the x coordinate of the tool tip and $\mathit{ToolTip}_{x^-}$ is the x coordinate of the tool tip at the previous state.

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:

$$\Box\neg(\mathit{ToolTip}_x = \mathit{ToolTip}_{x^-} \wedge \mathcal{X}(\mathit{ToolTip}_x = \mathit{ToolTip}_{x^-} \wedge \ldots \wedge \mathcal{X}(\mathit{ToolTip}_x = \mathit{ToolTip}_{x^-} \underbrace{)\ldots)}_{m-1} \wedge$$
$$\wedge \mathit{ToolTip}_y = \mathit{ToolTip}_{y^-} \wedge \mathcal{X}(\mathit{ToolTip}_y = \mathit{ToolTip}_{y^-} \wedge \ldots \wedge \mathcal{X}(\mathit{ToolTip}_y = \mathit{ToolTip}_{y^-} \underbrace{)\ldots)}_{m-1} \wedge$$
$$\wedge \mathit{ToolTip}_z = \mathit{ToolTip}_{z^-} \wedge \mathcal{X}(\mathit{ToolTip}_z = \mathit{ToolTip}_{z^-} \wedge \ldots \wedge \mathcal{X}(\mathit{ToolTip}_z = \mathit{ToolTip}_{z^-} \underbrace{)\ldots)}_{m-1}).$$

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:

$$\Box(\mathit{ToolTip}_x > 0 \wedge \mathit{ToolTip}_x < A \wedge \mathit{ToolTip}_y > 0 \wedge \mathit{ToolTip}_y < B).$$

Since the properties are described in LTL_f, we can use one of the existing software tools that will monitor this property in real time, for example the one discussed in [6, 13]. Run-time monitoring will be used to automatically alert the surgeon to potential problems during surgery, leading to increased safety of the procedure.

References

1. Bartocci, E., Falcone, Y. (eds.): Lectures on Runtime Verification - Introductory and Advanced Topics, Lecture Notes in Computer Science, vol. 10457. Springer (2018). https://doi.org/10.1007/978-3-319-75632-5, https://doi.org/10.1007/978-3-319-75632-5
2. Bresolin, D., Geretti, L., Muradore, R., Fiorini, P., Villa, T.: Formal verification applied to robotic surgery. Lecture Notes Control Inf Sci. **456**, 347–355 (2015). https://doi.org/doi:10.1007/978-3-319-10407-2˙40
3. Bresolin, D., Geretti, L., Muradore, R., Fiorini, P., Villa, T.: Formal verification of robotic surgery tasks by reachability analysis. Microprocess. Microsystems **39**(8), 836–842 (2015). https://doi.org/10.1016/j.micpro.2015.10.006, https://doi.org/10.1016/j.micpro.2015.10.006
4. Cerrito, S., Mayer, M.C., Praud, S.: First order linear temporal logic over finite time structures. In: Ganzinger, H., McAllester, D., Voronkov, A. (eds.) Logic for Programming and Automated Reasoning. pp. 62–76. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
5. Ferrando, A., Cardoso, R.C., Fisher, M., Ancona, D., Franceschini, L., Mascardi, V.: ROS-Monitoring: A runtime verification framework for ROS. In: Mohammad, A., Dong, X., Russo, M. (eds.) Towards Autonomous Robotic Systems - 21st Annual Conference, TAROS 2020. Lecture Notes in Computer Science, vol. 12228, pp. 387–399. Springer (2020)
6. Geatti, L., Montanari, A., Saccomanno, N.: 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). pp. 13–19 (2023)
7. Giacomo, G.D., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. pp. 854–860. IJCAI/AAAI (2013)
8. Hertz, B., Luppen, Z., Rozier, K.Y.: Integrating runtime verification into a sounding rocket control system. In: Dutle, A., Moscato, M.M., Titolo, L., Muñoz, C.A., Perez, I. (eds.) NASA Formal Methods - 13th International Symposium, NFM 2021, Virtual Event, May 24-28, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12673, pp. 151–159. Springer (2021). https://doi.org/10.1007/978-3-030-76384-8
9. Li, Z., Hutchinson, K., Alemzadeh, H.: Runtime detection of executional errors in robot-assisted surgery. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 3850–3856 (2022). https://doi.org/10.1109/ICRA46639.2022.9812034
10. Ruurda, J.P., van der Sluis, P.C., van der Horst, S., van Hilllegersberg, R.: Robot-assisted minimally invasive esophagectomy for esophageal cancer: A systematic review. J Surg Oncol. **112**, 257–265 (2015). https://doi.org/doi:10.1002/jso.23922
11. Saracino, A., Deguet, A., Staderini, F., Boushaki, M.N., Cianchi, F., Menciassi, A., Sinibaldi, E.: 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**(4), e1999 (2019)
12. Shao, Z., Xu, J., Stoyanov, D., Mazomenos, E.B., Jin, Y.: Think step by step: Chain-of-gesture prompting for error detection in robotic surgical videos (2024), https://arxiv.org/abs/2406.19217
13. Yamaguchi, T., Hoxha, B., Ničković, D.: RTAMT – Runtime Robustness Monitors with Application to CPS and Robotics. International Journal on Software Tools for Technology Transfer **26**, 79–99 (2024). https://doi.org/10.1007/s10009-023-00720-3