TIMED MODEL OF THE RADIATION THERAPY SYSTEM WITH RESPIRATORY MOTION COMPENSATION

Tomas KRILAVIČIUS *, KA LOK MAN **
*Vytautas Magnus university and Baltic Institute of Advanced Technologies; ** Xi'an Jiaotong-Liverpool University

***Abstract*:** The goal of the radiation therapy is to give as much dose as possible to the target volume and avoid giving any dose at all to a normal tissue. Despite the advances of the computer-based control current technology does not allow to compensate respiratory movement. It considerably restricts effectiveness of such treatment the case of lung cancer.
In this paper we present a work in progress, a timed model of radiation treatment system developed to analyze a potential set up for a system that compensates respiratory motion. We model the system with Uppaal, a tool for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.). The model is used to validate understanding of the model and selected scenarios.

***Keywords*:** formal methods, verification, medical systems, simulation

1. Introduction

The goal of the radiation therapy is to give as much dose as possible to the target volume of tissue and avoid giving any dose to a normal tissue. Advances of the computer-based control allow planning and performing accurate plans and treatments, however motion compensation during treatment remains a considerable problem. Different techniques to cope with such problem are discussed in [1]. Gating combined with external surrogates is overviewed in [2]. Other research models try predicting movement of the tumor, e.g. [3-5]. We, on the other hand, are interested in modeling hardware and software that should move precisely and fast to compensate respiratory movement. We use formal methods, because they provide means for rigorous modeling and analysis of diverse systems. Main reasons of the formal methods popularity are the following: formal modeling languages allow defining systems, including non deterministic behavior, unambiguously, and that allows applying rigorous reasoning about models, e.g. model checking, theorem proving and other specifically designed algorithms.
Quite a few techniques and tools were defined over the years, e.g. process algebras [5-10], timed automaton [11] and related Uppaal tool [12], hybrid automaton [13], see [14] for a wider overview. Successful application of formal techniques is reported in different areas, e.g. automotive [15], electronics [16], industrial devices control [17] and other.
In this paper we apply Uppaal tool [12] for the design and analysis of a radiation therapy system consisting of a HexaPOD couch with 6-degrees movement, a tracking camera, a marker (markers) and a controller. Uppaal is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types and other convenient constructions [12]. We extend model presented in [18] with time constraints and timed trajectories, and analyze selected properties of the systems.
In the second section we provide a description of the radiation treatment system. Then we introduce timed automaton and Uppaal and in the third section. In the fourth section we present and discuss an Uppaal model of the radiation treatment system and check some of its properties, and its applicability to further analysis. Future plans and conclusions close the paper.

2. Radiation Treatment System

There exists a plenitude of diverse radiation treatment systems, e.g. see [1]. We analyze a particular setup, depicted in Fig. 1, consists of the following components:

- **Patient Setup Couch** is used to position patient for the treatment, in our case the HexaPOD couch [19-20]
- **External Radiation Beam Source** usually produced by a *medical linear accelerator*, in short, *linac*. We omit it, because we analyze behavior of the couch and the controller only.
- **Tracking Device** provides information about the position of the patient. Different means and techniques

can be used to perform it, see [1] for the details. Our model a system is with a stereo camera, we omit it in this paper.

- **Controller** is a system that controls the treatment process, in our case the controller uses information provided by the treatment plan and the HexaPOD response to control it.

Fig. 1. Radiation Treatment System

3. Uppaal

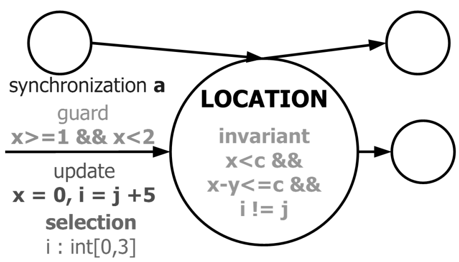

Fig. 2. Timed automaton.

Timed automaton [11] is a well know formalism for the modeling and analysis of sophisticated systems. We use timed automata based tool Uppaal [12], and therefore present a versions of timed automata adopted (and extended) in Uppaal.

Definition 1. Let $C = \{x, y, z, \dots\}$ be a *set of clocks* and $B(C)$ be the *set of clock restrictions* of the form $g, h \doteq x \bowtie c \mid x - y \bowtie c \mid g \wedge h$ with $x, y \in C,\ c \in \mathbb{N}$ *and* $\bowtie \in \{<, \leq, =, \geq, >\}$.

Definition 2. We will call *timed automaton* a finite directed graph over C and $B(C)$

$$A = (L, l_0, \mathcal{A}, E, I) \tag{1}$$

where L is a *finite set of locations*; $l_0 \epsilon L$ is the *initial location*; $\mathcal{A}$ is a *finite set of action names* and $E \subseteq L \times B(C) \times \mathcal{A} \times 2^C \times L$ is a *finite set of edges*, $I: L \to B(C)$ assigns *invariants* to locations.

We will write $l \xrightarrow{g,a,r} l'$ instead of $(l, g, a, r, l') \in E$. For such an edge, l is called the *source location* of the state, g is the *guard*, a is the *action*, r is the *set of clocks to be reset* and l' is the *target location*. Timed automata can be represented as in Fig. 2. Locations are drawn as nodes in the graph, and the initial location is usually marked with a double circle.

Definition 3. Let $A = (L, l_0, \mathcal{A}, E, I)$ be a timed automaton over C. We define the *timed transition system* generated by A as $T(A) = \left(S, Act, \overset{tr}{\rightarrow}\right)$ where:

- $S = L \times (C \longrightarrow \mathbb{R}_{\geq 0})$ is a *set of states* (l, v), where l is a location of the timed automaton and v is a clock valuation satisfying the invariant of l;
- $Act = \mathcal{A} \cup \mathbb{R}_{\geq 0}$ is the *set of labels*;
- Two types of transitions are defined:
  - *Action transitions* $(l, v) \overset{a}{\rightarrow} (l', v')$ such that exists an edge $(l \xrightarrow{g,a,r,} l') \in E$ where v satisfies g, v' satisfies $v[r]$ and v' satisfy $I(l')$,
  - Delay transitions $(l, v) \overset{d}{\rightarrow} (l', v')$ if $\forall d' \in [0, d] \Longrightarrow v + d$ satisfies $I(l)$.

Let v_0 denote the valuation with $v_0(x) = 0$ for all $x \in C$. If it satisfies the invariant of the initial location, we will call (l_0, v_0) the initial state of $T(A)$.

In Uppaal, timed automata are composed into a network, consisting of n timed automata $A_i = (L_i, l_i^0, \mathcal{A}, E_i, I_i), \mathrm{i} = 1 \dots \mathrm{n}$. Let $l = (l_1, \dots, l_n)$ be a location of the network, then invariants are composed using conjunction $I(\bar{l}) = \wedge_{i=0}^{n} I_i(l_i)$.

Definition 4. Let $A_i = (L_i, l_i^0, \mathcal{A}, C, E_i, Inv_i), \mathrm{i} = 1 \dots \mathrm{n}$ be a *network of* n *timed automata*. Let $\bar{l}_0 = (l_1^0, \dots, l_n^0)$ be the *initial location vector*. The semantics is defined as a *transition system* $(S, s_0, \rightarrow)$, where $S = (L_1 \times \dots \times L_n) \times \mathbb{R}^C$ is the *set of states*, $s_0 = (\bar{l_0}, v_0)$ is the *initial state* and transition relation combines 3 transitions types:

- *time flow* transitions $(\bar{l}, v) \overset{d}{\rightarrow} (\bar{l}, v + d)$, if $\forall d' \in [0, d]$ holds $v + d' \vDash Inv(\bar{l})$;
- *discrete transitions*
  - *synchronized* $\left((l_1 \dots, l_i, \dots, l_j, \dots, l_n), v\right) \overset{\tau}{\rightarrow} \left((l_1 \dots, l'_i, \dots, l'_j, \dots, l_n), v'\right)$ if $\exists i \neq j$, $\exists \left(l_i \xrightarrow{a!, g_i, r_i} l'_i\right) \in E_i, \exists \left(l_j \xrightarrow{a?, g_j, r_j} l'_j\right) \in E_j,\ v \vdash g_i \wedge g_j,\ v' \vdash v[r_i \cup r_j]$ and $v' \vdash I_i(l_i) \wedge I_i(l_i) \wedge \wedge_{k \neq i,j} I_k(l_k)$,
  - *asynchronous* $\left((l_1 \dots, l_i, \dots, l_n), v\right) \overset{\tau}{\rightarrow} \left((l_1 \dots, l'_i, \dots, l_n), v'\right)$ if $\exists \left(l_i \xrightarrow{\tau, g, r} l'_i\right) \in E_i,\ v \vdash g,\ v' \vdash v[r]$ and $v' \vdash Inv_i(l_i) \wedge \wedge_{k \neq i} Inv_k(l_k)$.

Uppaal is based on the theory of timed automata; however it's modeling language offers additional features such as bounded integer variables, urgency, and more [12]. Properties to be verified are specified using a subset of CTL (*computation tree logic*) [21]:

- **A[] p**: *invariant* - property p always holds in all paths;
- **A<> p**: *eventually* - property p holds in all paths at some moment;
- **E<> p**: *possibly* - property p eventually holds at some state, at least in one path;
- **E[] p**: *potentially always* - property p eventually holds from some state, at least in one path;

- **p-> q**: *leads to* - whenever *p* holds eventually *q* will hold as well;
- **deadlock**: true, if deadlock is reachable;
- **P.state p**: property p holds in the selected state.

4. Timed Model of the Radiation Treatment System

We present a work in progress, a simplified version of the radiation treatment system defined in Sect. 2. We extend model presented in [18] with timing aspects. The model is used to explore HexaPOD behavior, therefore we ignore the Tracker and the model consists of the following components:

- **Controller** sends commands to HexaPOD based on its state, a treatment plan (in this case, trajectory) and the input from the tracking system, i.e. stereo camera[1];
- **HexaPOD** moves according to its physical limitations and the commands sent from the Controller via the **HexaPOD Buffer** that models asynchronous communication and latency between the controller and the HexaPOD.

4.1. HexaPOD and HexaPOD Buffer

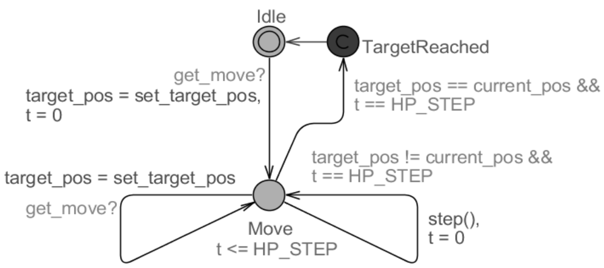

Fig. 3. HexaPOD.

An Uppaal model of the HexaPOD is depicted in Fig. 3. We model it as a one point-device with a discrete movement in three - *x, y* and *z* directions. We abstract from the acceleration and rotation. Such a simplified model still allows investigating an impact of the latency and the general design of HexaPOD control. The automaton consists of three locations:

1. **Idle**: HexaPOD waits for a command **move_to**. With this action it receives a target, and changes to **Move** location.
2. **Move:** HexaPOD stepwise moves towards the target, taking steps in the predefined direction of the predefined length at a constant speed. After each step it checks for a new target, and updates the current one, if necessary. When the target is reached, it changes to **TargetReached** location.
3. **TargetReached** is a *committed* location (a special type of location, which should be left at the next step), which is used for diagnostic reasons, see Sect.

HexaPODBuffer models asynchronous communication and latency. It consists of three locations. **Empty** location denotes an empty buffer; it idles until the **move_to** Command from the Controller and then changes to **Latency** location. **Latency** location is used to model delays in the system, i.e. after receiving the new target the buffer delays before making it available to the HexaPOD. However, the new target can be provided to the buffer anytime. **Ready** location denotes readiness of the buffer: the target can be acquired by the HexaPOD using **get_move** action.

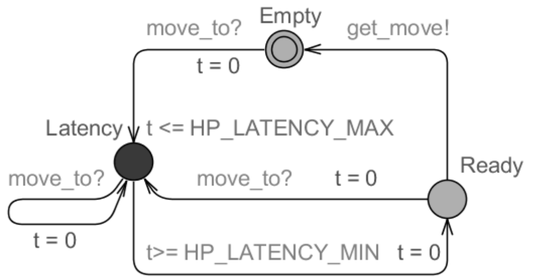

Fig. 4. HexaPOD Buffer.

4.2. Controller

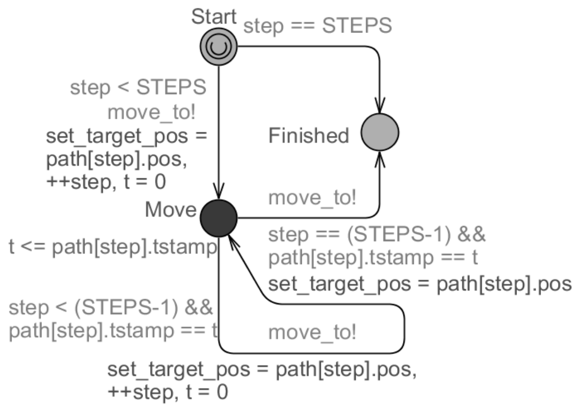

Fig. 5. Controller.

The controller just commands to the HexaPOD. It consists of three locations: **Start, Move** and **Finished**, where the first and the third denote the beginning and the end of control, correspondingly. In the **Move** location, the system sends control inputs at the defined time moments, all provided by an array.

4.3 Queries

We use the following queries to generate diagnostic traces as well as to validate the model.

1. **E<> Controller.Finished**, **A<> Controller.Finished** checks, can Controller send all control commands. The first one shows existence at least one path that leads to the Finished state of the Controller, while the last proves that in all paths eventually such state is reached.
2. **A<> Controller.Finished and Controller.step == Controller.STEPS-1** – if Controller is in the state Finished then it has sent all the commands.
3. **E<> HexaPOD.TargetReached and HexaPOD. current_pos == Controller.path[Controller.STEPS-1].pos** – HexaPOD has reached the target and it coincides with the defined target.
4. **A<> HexaPOD.TargetReached and HexaPOD. current_pos == Controller.path[Controller.STEPS-1].pos** – HexaPOD in all paths eventually reaches the

[1] We ignore the tracking device in this paper.

target position, and it arrives at a proper position as well.

Provided queries allows validating consistency of the systems and producing diagnostic traces, that can be compared with expected traces and behavioral properties of the HexaPOD estimated.

5. Conclusions

We have presented a work in progress – a timed Uppaal model of the radiation treatment system. The model can be used to analyze functional properties of the HexaPOD and estimate requirements for couch that would be able to follow required, e.g. breathing cycle, trajectories. It shows limitations of the approach as well: trajectories can be only approximated, time scales are to different, distances modeling precision is insufficient.

Our future plans include

1. Extensions of the Uppaal model:
   i. model of HexaPOD with rotation, acceleration and more precise velocity;
   ii. model of the targeting component;
   iii. implementation of the different control approaches.
2. Continuous model of the HexaPOD, that would allow increase the precision of the timed model and would generate discrete control and testing paths for it.
3. Formal hybrid model, using Behavioral Hybrid Process Calculus [8, 9] or a semi-formal control model in OpenModelica [22].
4. Combination of the real respiratory movement trajectories and (formal) model to investigate the systems' adequacy to compensate it.

Acknowledgements. We thank Kaiyu Wan, D. Hughes and K. Lee for the collaboration on the preliminary model [18].

7. References

1. Keall, P. J., Mageras, G. S.; Balter, et al. The management of respiratory motion in radiation oncology report of AAPM Task Group 76. Medical Physics, AAPM, 2006, 33, p. 3874-3900.
2. Berbeco, R. I., Nishioka, S., Shirato, H., Chen, G. T. Y. and Jiang, S. B. Residual motion of lung tumours in gated radiotherapy with external respiratory surrogates. Physics in Medicine and Biology, 2005, 50, p. 3655-3667.
3. Ruan, D., Fessler, J. A. and Balter, J. M. Real-time prediction of respiratory motion based on local regression methods. Physics in Medicine and Biology, 2007, 52, p. 7137- 7152.
4. Kalet, A., Sandison, G., Wu, H. and Schmitz, R. A state-based probabilistic model for tumor respiratory motion prediction. Physics in Medicine and Biology, 2010, 55, 7615-7631.
5. Wu, H., Sharp, G. C., Salzberg, B., Kaeli, D., Shirato, H. and Jiang, S. B.A finite state model for respiratory motion analysis in image guided radiation therapy. Physics in Medicine and Biology, 2004, 49, p. 5357-5372.
6. Milner R. Communication and Concurrency. Prentice-Hall, Inc., 1989. 272 p.
7. Hoare C.A.R. Communicating Sequential Processes. Prentice-Hall, Inc., 1985. 256 p.
8. Brinksma E., Krilavičius T. and Usenko Y.S. Process Algebraic Approach to Hybrid Systems. Proceedings of the 16th IFAC World Congress, Prague, 2005, 6 p.
9. Krilavičius T. Hybrid Techniques for Hybrid Systems. PhD thesis, University of Twente, Enschede, 2006. 192 p.
10. van Beek D., Man K., Reniers M., Rooda J. and Schiffelers R. Syntax and consistent equation semantics of hybrid chi. Tech. Rep., Tech. Univ. of Eindhoven (TU/e), Eindhoven, 2004. 119 p.
11. Alur R. and Dill D. The theory of timed automata. Proceedings of Real-Time: Theory in Practice, Springer, 600, LNCS, 1994, p. 45–73.
12. Behrmann G., David A. and Larsen, K.G. A Tutorial on Uppaal. Proceedings of the 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT'04). LNCS 3185, 2004, p. 200—236.
13. Alur R., Courcoubetis C., Henzinger T. and Ho P.-H. Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems. Hybrid Systems, vol. 736, 1993, p. 209-229.
14. Bowen J.P. and Hinchey. Ten Commandments of Formal Methods…Ten Years Later, Computer, vol. 39, no. ,1 IEEE, January, 2006, p. 40-48.
15. Gebremichael B., Krilavičius T. and Usenko Y.S. A formal model of a car periphery supervision system in UPPAAL. Proceedings of Workshop on Discrete Event Systems (WODES'04), September 22-24, 2004, Reims, France, pp. 433-438.
16. Man K.L. and Krilavičius T. Case Studies with Process Analysis Toolkit (PAT). International SoC Design Conference, BEXCO, Busan, 2009, 4 p.
17. Krilavičius T. And Miliukas V. Functional Modeling and Analysis of Distributed Lifting System. The 5th International Conference on Electrical and Control Technologies (ECT 2010), 2010. 6 p.
18. Krilavičius T., Man, K.L., Kaiyu Wan, Hughes D. and Lee, K. Modeling and Analysis of Radiation Therapy System with Respiratory Compensation using Uppaal. The 9th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2011), Busan, 2010, 6 p. (submitted).
19. Chung H., Jin H., Suh T., Palta J. and Kim, S. Characterization of a commercial add-on couch, HexaPOD™ 6D Robotic Treatment CouchTOP. World Congress on Medical Physics and Biomedical Engineering 2007, 14, p. 1945-1947.
20. Elekta Synergy® S with HexaPOD™, 2006, 4p.
21. Clarke E.M., Grumberg O. and Peled D.A. Model Checking. The MIT Press, 2001. 314 p.
22. Open Modelica. http://www.openmodelica.org/. 2011, last visited: 2011 03 07.