724. Simulation of the radiation therapy system for respiratory movement compensation

Tomas Krilavičius[1], Daiva Vitkutė-Adžgauskienė[2], Kęstutis Šidlauskas[3]
Vytautas Magnus University, Kaunas, Lithuania
E-mail: [1]*T.Krilavicius@if.vdu.lt*, [2]*D.Vitkute-Adzgauskienė@if.vdu.lt*, [3]*K.Sidlauskas@if.vdu.lt*

(Received 11 September 2011; accepted 14 February 2012)

Abstract. 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 healthy tissue. Advances of the digital control allow performing accurate plans and treatments. Unfortunately, motion compensation during the treatment remains a considerable problem. Currently, combination of the different techniques, such as gating (restricting movement of patient) and periodic emission are used to avoid damaging healthy tissue We are interested in systems that completely compensate respiratory movement (up to certain limit) and start by investigating adequacy of the existing hardware and software platform.

We model a radiation therapy system consisting of a HexaPOD couch with 6-degrees movement, a tracking camera, a marker (markers) and a controller. Formal un-timed and timed models were defined, analyzed and found to be insufficient to completely determine adequacy of the system to compensate respiratory motion. We define one-dimensional hybrid model of the system using Open Modelica tool and investigate the model with simple tumor movement trajectories, and based on the results we sketch further development directions.

Keywords: simulation, radiation treatment, quality assurance.

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 healthy tissue. Advances of the computer-based control allow planning and performing accurate plans and treatments; however motion compensation during the treatment remains a considerable problem. Overview of different techniques is given in [1], gating with external surrogates examined in [2], while [3-5] analyze models for predicting movement of the tumor. We, on the other hand, are interested in modeling hardware and software capable of moving precisely and fast to compensate respiratory movement.

We model a radiation therapy system consisting of a HexaPOD couch [6, 7] with 6-degrees movement (3 translational + 3 rotational), a tracking camera, a marker (markers) and a controller. In [8] formal modeling tool Uppaal [9] is used to investigate un-timed behavior of the system. Uppaal-based timed model is provided in [10]. However, both models were found to be insufficient to completely determine adequacy of the system to compensate respiratory motion. In this paper we describe a selected radiation treatment system setup. Then we define a one-dimensional hybrid model of the system using Open Modelica tool [11], an open-source Modelica-based [12] modeling and simulation environment and investigate the model with simple trajectories. We finish paper by sketching further development directions.

Radiation Treatment System

Different radiation treatment systems are used, e. g. see [1]. We analyze a particular setup, depicted in Fig. 1, based on the system produced by Rubedo Systems (Rubedo Systems iGuide, http://rubedo.lt/Sprendimai/iGUIDELT.aspx) and Elekta AB (Elekta Oncology Products, www.elekta.com/healthcare_international_elekta_oncology.php), consisting of the following components:

1. **Patient Setup Couch** positions patient for the treatment, in our case it is HexaPOD couch [6, 7]. The couch has 6 degrees of freedom: 3 directions of rotational movement (superior-inferior (SI, i. e. head-legs), left-right (LR) and anterior-posterior (AP, i. e. breast-back)) and 3 directions of translational movement.
2. **External Radiation Beam Source** is omitted, because it is considered to be static.
3. **Tracking Device** provides the position of the marker or tumor. Different techniques can be used to perform it, see [1]. We model a system is with a stereo camera, however in this model it is a device that just samples given trajectory with a preset frequency.
4. **Controller** guides the treatment process. In our model it attempts to direct the HexaPOD in such a way that it follows the position provided by the camera as closely as possible.

Fig. 1. Radiation Treatment System

One Dimensional Model of the System

```
model SinusoidalCamera "Camera samples marker position with preset frequency"
  import Modelica.Constants.*;
  import Modelica.Math.*;
  import SI=Modelica.SIunits;
           SI.Position y ;      // position of the object under observation, m
  discrete SI.Position target; // position of the target at sampling time, m
protected
  constant SI.Period PERIOD = 1/20;       // sampling period, s
  // sine wave modifiers
  constant Real AMPLITUDE         = 1.5e-3;    // amplitude coefficient
  constant Real SHIFT             = AMPLITUDE; // shift, m
  constant SI.Frequency FREQUENCY = 1/3;       // frequency, Hz
equation
  y = AMPLITUDE * Modelica.Math.sin(2*pi*FREQUENCY*time) + SHIFT;
  when sample(0, PERIOD) then
    target = y;
  end when;
end SinusoidalCamera;
```

Fig. 2. Camera in OpenModelica

Our goal is to investigate an adequacy of the system to compensate respiratory motion, especially for lungs tumor treatment. We define one-dimensional model to investigate feasibility of the existing hardware and software platform and applicability of the tool for the task. Tumors usually move in all three dimension, therefore our future plans involve extending the model, however in some cases [13] even one dimension (SI) suffices. In our case, we use anterior-posterior (AP) dimension, because it seems to be more important in the case of lungs movement, however, the next iteration of the model will be extended to include all three dimensions: superior-inferior (SI), left-right (LR) and anterior-posterior (AP).

We have chosen Modelica language [12] and Open Modelica tool [11] for modeling and analysis of the system. Modelica is a free object-oriented modeling language with a textual definition to describe physical systems in a convenient way by differential, algebraic and discrete equations. It is well supported by different tools, but we have chosen an open-source free alternative, Open Modelica [11] to assess its maturity and its suitability for the modeling of mixed software-hardware systems. We use OMNotebook (OpenModelica Notebook (OMNotebook), www.openmodelica.org/index.php/developer/tools/137) to model the system, because it allows literate programming [14], i. e. the model and documentation of the model are merged together, and it is possible to read the documentation and interactively simulate the model.

We use a modular approach, i.e. model separate components and then connect them via communication channel, in Modelica case – connectors. The system consists of the three components listed below connected in to a system by HexaPODD1_Controller_Camera model.

Camera samples input (movement of the tumor or marker) with a preset frequency and makes it available for the controller. In the current model it samples simulation of the respiratory movement provided as a continuous function, but in the further investigation recorded movement of tumor and markers will be used. Sampling introduces delays in to the system. Camera in OpenModelica samples input trajectory, in this case sine function modified to resemble respiratory movement is depicted Fig. 2. Line

```
y = AMPLITUDE * Modelica.Math.sin(2*pi*FREQUENCY*time) + SHIFT;
```

can be substituted by any chosen function or trajectory.

Controller samples output of the camera observations at preset periodicity and sets new targets for HexaPOD. Again, sampling introduces delays, and consequently an error. Controller is depicted in Fig. 3.

```
model Controller
  import SI=Modelica.SIunits;
  discrete SI.Position position;    // position from camera, m
  discrete SI.Position target;      // position set by controller, m
protected
  constant SI.Period PERIOD = 1/15; // sampling period, s
equation
  when sample(0, PERIOD) then
    target = pre(position);
  end when;
end Controller;
```

Fig. 3. Controller in OpenModelica

HexaPOD moves according its physical characteristics towards the target provided by Controller. Target position depends on the sampling frequencies of Camera and Controller. Additional delays can be introduced by HexaPOD latency; however it is ignored in the current model. HexaPOD model is depicted in Fig. 4.

HexaPOD exhibits hybrid behavior – it moves up, down and stops when reaches the target. Therefore, it should detect change of direction as well as time, when the target is reached. Direction of the movement is denoted by a boolean variable *up* that flips, when one of the following conditions is met:

1. `up and x > target+delta`, i.e. *up* is true and HexaPOD is above *target* + *delta*, where *delta* is a workaround to overcome chattering that occurs due to numerical solvers errors. Chattering is a phenomenon which occurs when just after switching the conditions become valid again and another switching occurs immediately. It is referred to as Zeno behavior [15] as well.
2. `not up and x < target-delta`, i.e. *up* is false and HexaPOD is below the *target-delta*.

Moreover, when HexaPOD reaches the target, it stops moving. It is defined by the *if statement* that checks Hexapods' movement direction and its vicinity to target.

Such definitions are not very elegant, but they allow defining ODE and DAE equations in such a way that solvers are able to detect the switching moment, then reinitialize and continue simulation in with different equations and initial conditions.

```
model HexaPODD1  "Model of one-dimensional HexaPOD with constant velocity"
  import Modelica.Math.*;
  import SI=Modelica.SIunits;
           SI.Position x (start=0);     // position of HexaPOD, m
  discrete SI.Position target;          // target, m
protected
  constant SI.Velocity velocity = 16e-3; // max velocity in m/s
           Boolean up (start=true);
  constant SI.Distance delta = 1e-5;
equation
  if up then
    if x < target then
      der(x) = velocity;
    else
      der(x) = 0;
    end if;
  else
    if x > target then
      der(x) = -velocity;
    else
      der(x) = 0;
    end if;
  end if;
  when {(up and (x > target+delta)), ((not up) and (x < target-delta))} then
   up = x < target;
  end when;
end HexaPODD1;
```

Fig. 4. Model of HexaPOD in OpenModelica

```
model HexaPODD1_Controller_Camera
  HexaPODD1 hexaPOD;
  SinusoidalCamera camera;
  Controller controller;
equation
  connect(controller.position, camera.target);
  connect(controller.target, hexaPOD.target);
end HexaPODD1_Controller_Camera;
```

Fig. 5. Aggregate Model of the System in OpenModelica

Final component of the model, HexaPODD1_Controller_Camera, just instantiates all the models and connects them. It is depicted in Fig. 5.

Simulation Results

We simulate model using sine trajectory modified to be similar to respiratory movement, i.e. $1.5 \cdot 10^{-3}$ m amplitude, the same shift, 1/3 frequency and selected parameters: 20 Hz camera sampling, 15Hz controller sampling, $16 \cdot 10^{-3}$ m/s or 0 m/s velocity. Simulation is performed using standard solver parameters. Simulation results are depicted in Fig. 6. HexaPOD tries following sine trajectory however delays introduced by the sampling produce a noticeable error, depicted in Fig. 7.

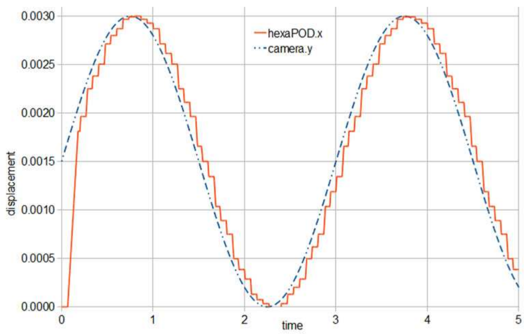

Fig. 6. Simulation of HexaPOD Movement

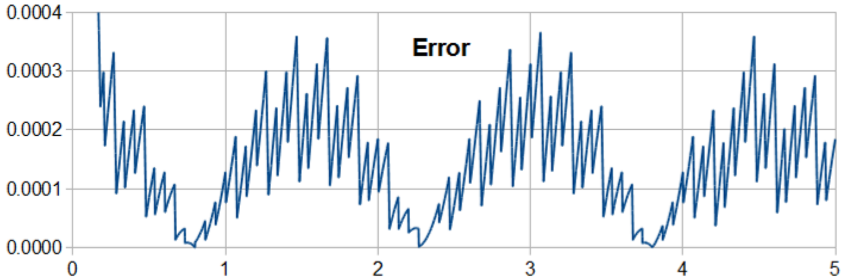

Fig. 7. Simulation Error: real trajectory – HexaPOD position

Error is defined as a distance between trajectory and HexaPOD position, we depict absolute value, and ignore initialization period error, i.e. initial positioning of HexaPOD. The error is quite small it hardly reaches $0.3 \cdot 10^{-3}$ m. However, acceleration and HexaPOD latency are neglected therefore the model should be made more precise to re-evaluate such results. Moreover, respiratory movement is more chaotic than used trajectory, therefore simulation with real trajectories maybe not so excelling.

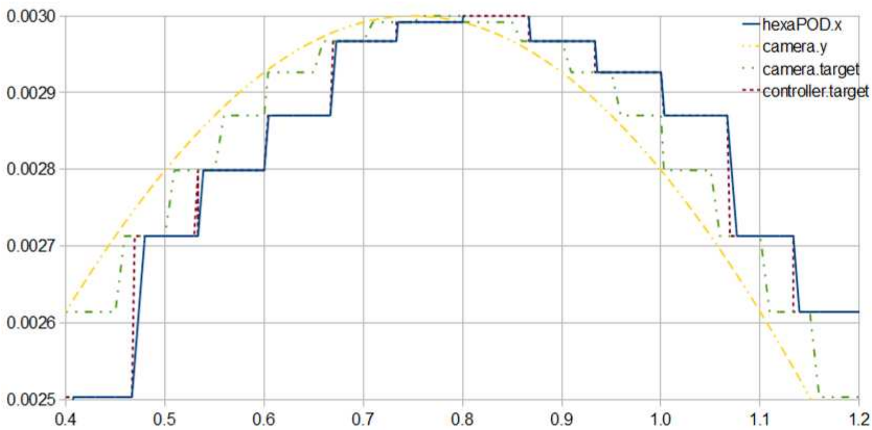

Fig. 8. Simulation results with intermediate variables

From Fig. 8 it is easy to see how delays accumulate when Controller samples the target just before camera. Therefore, for Controller in some cases it could be better to acquire value from Camera later, i.e. sampling frequency of Controller and Camera should be synchronized in some manner. Moreover, it shows simulation error produced by OpenModelica, i. e. value of *camera.target* should change at the same time point, but due to the simulation step size there is a small time interval between these values. When simulated separately, the step size is adjusted and simulation results are precise.

Simulation results hint that some prediction of the movement would improve control of HexaPOD as well. Moreover, we expect that some delays can be overcome changing sampling frequencies.

Conclusions and Future Plans

We have presented a work in progress, a hybrid model of radiation treatment system for the respiratory motion compensation analysis. The tool seems to be adequate for the task, however certain problems were noticed: OMNotebook quite often becomes unresponsive, but restarting it helps. Moreover, plotting functionality is not sufficient for producing display ready plots therefore simulation data was generated in comma-separated values and Open Office Calc was used to generate figures for this paper.

Described model allows analyzing an influence of delays introduced by camera and controller sampling to the movement of HexaPOD. Moreover, it allows experimenting with different delays and trajectories. It shows quite promising results with artificial trajectories. However, real respiratory trajectories are more chaotic. Moreover, acceleration and latency of HexaPOD are neglected in the current study. Therefore the model should be extended to get more realistic results.

1. Three-dimensional movement, latency and acceleration (actually, initial tests with acceleration were made, but more experiments with HexaPOD should be performed to determine it) should be added to HexaPOD model.
2. Realistic trajectories of tumor and/or marker movement should be recorded and provided to camera.
3. Latency of controller should be estimated, and both, upper and lower bounds determined.

Moreover, we plan to compare the model with an experimental layout and adjust it.

References

[1] **Keall P. J., Mageras G. S., Balter J. M.** The Management of Respiratory Motion in Radiation Oncology. Report of AAPM Task Group 76. Med. Phys., 2006, 33, p. 3874-3900.

[2] **Berbeco R. I., Nishioka S., Shirato H., et al.** Residual Motion of Lung Tumours in Gated Radiotherapy with External Respiratory Surrogates. Phys. in Med. and Bio., 2005, 50, p. 3655-3667.

[3] **Ruan D., Fessler J. A., Balter J. M.** Real-Time Prediction of Respiratory Motion Based on Local Regression Methods. Phys. in Med. and Bio., 2007, 52, p. 7137- 7152.

[4] **Kalet A., Sandison G., Wu H., Schmitz R.** A State-Based Probabilistic Model for Tumor Respiratory Motion Prediction. Phys. in Med. and Bio., 2010, 55, p. 7615-7631.

[5] **Wu H., Sharp G. C., Salzberg B., et al.** A Finite State Model for Respiratory Motion Analysis in Image Guided Radiation Therapy. Phys. in Med. and Bio., 2004, 49, p. 5357-5372.

[6] Elekta Synergy® S with HexaPOD™, 2006, p. 4.

[7] **Chung H., Jin H., Suh T., Palta J., Kim S.** Characterization of a Commercial Add-on Couch, HexaPOD™ 6D Robotic Treatment CouchTOP. World Congress on Med. Phys. and Biomedical Engineering, 2007, 14, p. 1945-1947.

[8] **Krilavičius T., Man K. L., Kaiyu Wan, et al.** Modeling and Analysis of Radiation Therapy System with Respiratory Compensation Using Uppaal. The 9th IEEE Int. Symp. on Parallel and Distributed Processing with Applications (ISPA 2011), 2011, 6 p.

[9] **Behrmann G., David A., Larsen K. G.** Tutorial on Uppaal. Proc. of the 4th Int. School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM-RT'04), LNCS 3185, 2004, p. 200-236.

[10] **Krilavičius T., Man K. L.** Timed Model of Radiation Therapy Treatment System with Respiratory Motion Compensation. The 6th Int. Conf. on Electrical and Control Technologies (ECT 2011), 2011, p. 4.

[11] Open Modelica: http://www.openmodelica.org/. 2011, last visited: 2011 03 07.

[12] Modelica – A Unified Object-Oriented Language for Physical Systems Modeling: Language Specification, Version 3.2, 2010 03 24: https://www.modelica.org/documents/ModelicaSpec32.pdf, p. 250.

[13] **Wu J.-K., Chen J.-C,, Cheng J., et al.** Model Analysis of Respiration-Related Dosimetric Change During Radiotherapy. Biomed. Eng.: App., Basis and Comm., 14:1, 2002, p. 35-39.

[14] Wikipedia contributors: Literate programming. Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=Literate_programming&oldid=425325152.

[15] **Krilavičius T.** Hybrid Techniques for Hybrid Systems. University of Twente, 2006, p. 207.