A Supervisory Control Synthesis Case Study: The Antenna Control System*

M. Barbeau, M. Frappier, F. Kabanza, and R. St-Denis
Département de mathématiques et d'informatique
Université de Sherbrooke
Sherbrooke, Québec CANADA J1K 2R1

Abstract

We devised an algorithm for deriving controllers for timed discrete-event systems with nonterminating behavior modeled by timed transition graphs and control requirements expressed by Metric Temporal Logic (MTL) formulas. This algorithm has several interesting features. Firstly, it simultaneously handles the issues of controllability, safety, liveness, and real time in a single framework. Secondly, time and space complexity of the algorithm is reduced thanks to ingenious search and representation techniques. This algorithm has a general character and is implemented in a system called Temporal Controller Synthesis Tool (TCST).

In our research, there is a strong commitment to the application of theoretical research results. We validated our algorithm through case studies. This is very important for assessing its realism. This paper presents the application of our algorithm to the derivation of a controller for an Antenna Rotor Control System (ARCS). The ARCS is responsible for orienting antennas in the direction of a telecommunications satellite. The paper presents modeling of the rotor system with timed transition graphs, specification of the constraints with MTL, and a controller derived using TCST.

1 Introduction

Synthesis of controllers is defined as a process aiming at systematically deriving a model of the behavior of a supervisor given a specification of control requirements and a model

*The research described in this paper was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

of the perception of the behavior of a process. Assessment of the suitability of synthesis approaches on real control problems is important in order to uncover flaws and guide future research activities. This paper presents the development of a synthesis case study using an algorithm handling liveness, safety, and real-time control constraints. The control problem consists of deriving the kernel of an Antenna Rotor Control System (ARCS).

The ARCS is responsible for maintaining the orientation of antennas in the direction of a moving telecommunications satellite. The direction is expressed in terms of an azimuth and an elevation (both in degrees). Its central component is an Antenna Direction Controller (ADC) responsible of deciding when to start/stop rotors and, whenever they are running, the direction of their movement. For this purpose, the ADC is connected to specialized movement sensing and control processes. There are separate sensing and control processes for azimuth and elevation. The sensing elements inform about the position of the antennas relative to an azimuth target and an elevation target. The rotor controllers are responsible for the timing details involved in switching on and off motors of the rotor in one direction or the other. Finally, the ADC is also responsible for synchronizing the operation of the controllers with the operation of a larger satellite tracking process (the antennas are pictured in Figure 1). Rotors appear at the junction of the vertical and horizontal masts.

The architecture of ARCS is illustrated in Figure 2 (this design is inspired from a design of a traffic light control system that appears in [5]). A solution for the ADC is developed using a synthesis algorithm described in References [1] and [2]. This algorithm allows synthesis from control requirements that include safety, liveness, and real-time properties. The control requirements are expressed by Metric Temporal Logic (MTL) formulas and the nonterminating behavior of the system is modeled by a timed transition graph. Events have time duration and states are labeled by propositional symbols. This algorithm is also quite close to one described in [3], except that the latter is formulated in an artificial intelligence planning paradigm.

The rest of this paper is structured as follows. Section 2 presents the model of the perception of the process and the control requirements. Section 3 discusses a solution. We conclude with Section 4.

2 Modeling of ADC

We model a process as a timed transition graph $G = (X, \mathcal{P}, \lambda, A, \tau, \xi, x_0)$, where X is a finite set of states; $\mathcal{P}$ is a finite set of propositional symbols; $\lambda : X \to 2^{\mathcal{P}}$ is a labeling function that assigns to each state the set of propositional symbols true at that state; A is a finite set of actions; $\tau : A \to \mathbb{R}^+$ is a time duration function such that $\tau(a) > 0$

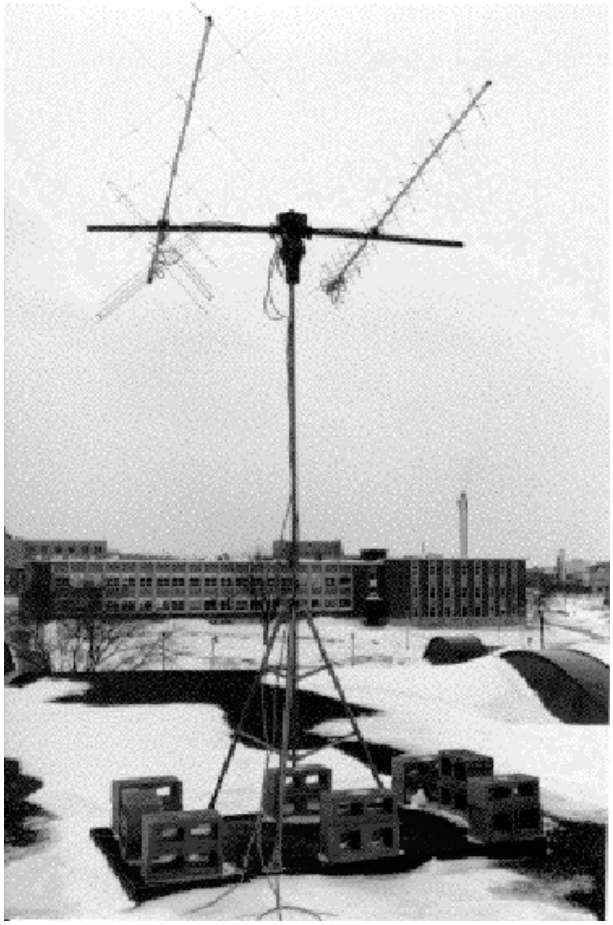

Figure 1: Antennas of a satellite tracking system

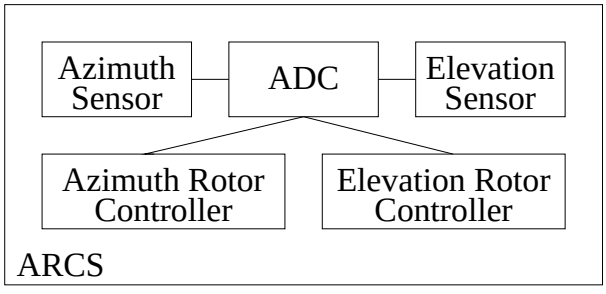

Figure 2: Architecture

for all $a \in A$; $\xi : X \times A \rightarrow X$ is a transition function; and $x_0 \in X$ is an initial state. The set A is partitioned into the sets A_c and A_{uc} denoting the set of *controllable* actions and set of *uncontrollable* actions, respectively. In the sequel, we detail only the elevation behavior and control, the azimuth behavior and control are analogous.

The ADC manipulates two control variables : *current* and *target*. The former represents the current position of antennas, whereas the latter represents its targeted position. An antenna is considered on target when the distance between *current* and *target* is less or equal than a constant d.

The domain of *current* and *target* is continuous, from zero to 180 degrees. However, we only reason about an abstract model of the continuous behavior in which only the relations between the variables *current* and *target* are relevant. Propositions *Low*, *Good*, and *High* refer to the current position of the antenna with respect to the target and they hold when the conditions $(target - current > d)$, $|target - current| \leq d$, or $(current - target > d)$ are respectively true. Initially, the relation between *target* and *current* is unknown which is conveyed by the proposition *Unknown*.

The propositions *Idle*, *Moving Down*, and *Moving Up* refer to the state of the elevation rotor. The value of $\lambda(x_0)$ is $\{Idle, Unknown\}$, i.e., the propositions true in the initial state x_0.

For the ADC, the set of propositional symbols $\mathcal{P}$ contains *Low*, *Good*, *High*, *Unknown*, *Idle*, *Moving Down*, and *Moving Up*.

The transition function ξ is often better represented by a diagram. We use Petri nets diagrams in which circles represent propositions and rectangles represent transitions. Transitions are connected to propositions. Incoming propositions are retracted, outgoing propositions are asserted, and propositions connected with arrows at both sides are tested by the transition. In our model, every action has a duration of one time unit.

The diagrams for the ADC are quite simply described. Target specification can be performed solely when propositions *Idle* and *Unknown* are true (i.e., the uncontrollable actions *Setpos Low*, *Setpos Good*, or *Setpos High*, on Figure 3, is executed). This has three possible outcomes: the current antenna position, relatively to the target, is set to either *Low*, *Good*, or *High*. On the other hand, when the current antenna position is good and the rotor is idle, the target position can be deleted causing *Good* to be retracted and *Unknown* to be asserted (i.e., the uncontrollable action *Deltpos* is executed). The transitions modeling the activation and deactivation of the elevation rotor are pictured in Figure 4. All these actions are controllable. The transitions of Figure 5 model the current antenna position monitoring, while the rotor is active. All these actions are controllable except action *Wait* which models inactivity.

The control requirements are specified with Metric Temporal Logic (MTL) [4], in which time constraints are associated with modal operators. This allows the specification

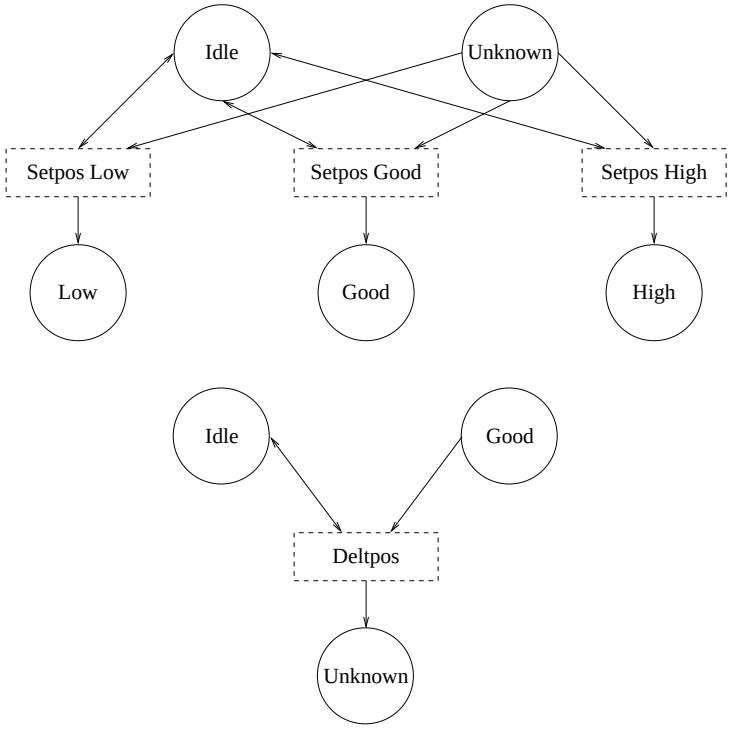

Figure 3: Setting and deleting target position

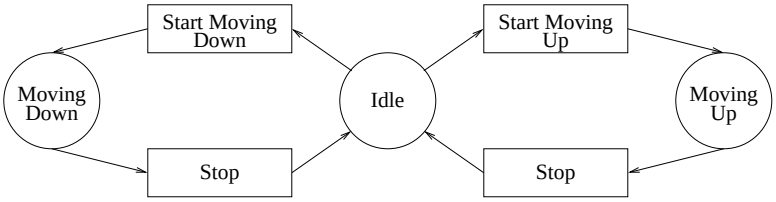

Figure 4: Rotor activation and deactivation transitions

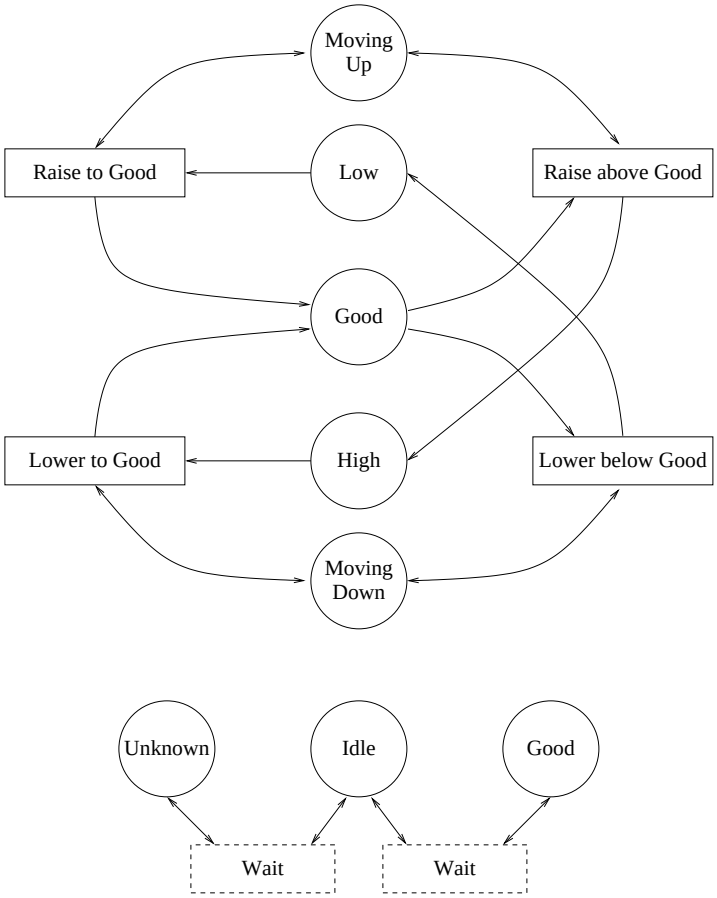

Figure 5: Monitoring transitions

of various properties such as "eventually, within t time units, property p will be satisfied" or "property p must always be satisfied after t time units." The following formulas specify the elevation constraints.

When a target is specified (i.e., current antenna position is unknown), the antenna must eventually reach the good position.

$$\Box_{\geq 0}[\neg unknown \rightarrow \Diamond_{\geq 0} Good] \tag{C1}$$

Whenever the motor is running and the antenna is at the good position, the motor must be stopped.

$$\Box_{\geq 0}[((Moving\ Down \vee Moving\ Up) \wedge Good) \rightarrow \bigcirc_{\geq 0}(Idle \wedge Good)] \tag{C2}$$

If the motor is idle and the antenna is to high, then the motor must be started in the down direction.

$$\Box_{\geq 0}[(Idle \wedge High) \rightarrow \bigcirc_{\geq 0} Moving\ Down] \tag{C3}$$

If the motor is running in the down direction and the antenna is to high, then the motor must keep running in the down direction.

$$\Box_{\geq 0}[(Moving\ Down \wedge High) \rightarrow \bigcirc_{\geq 0} Moving\ Down] \tag{C4}$$

If the motor is idle and the antenna is to low, then the motor must be started immediately in the up direction.

$$\Box_{\geq 0}[(Idle \wedge Low) \rightarrow \bigcirc_{\geq 0} Moving\ Up] \tag{C5}$$

If the motor is running in the up direction and the antenna is to low, then the motor must keep running in the up direction.

$$\Box_{\geq 0}[(Moving\ Up \wedge Low) \rightarrow \bigcirc_{\geq 0} Moving\ Up] \tag{C6}$$

If the antenna position is good, wait or do nothing.

$$\Box_{\geq 0}[Good \rightarrow \bigcirc_{\geq 0}(Idle \vee Good)] \tag{C7}$$

3 Synthesis of a Solution

We discuss synthesis of the controller ADC using an algorithm presented in References [1] and [2]. This algorithm produces a controller by using a forward search technique, combined with incremental model checking. More specifically, this is a combination of three main operations:

- Incremental exploration of the global state space of the interleaved processes, while verifying the MTL formulas over trajectories (sequences of states) to detect bad states (violations of safety properties) or bad cycles (violations of liveness properties). The result is a graph of global states labeled by MTL formulas.

- Use of a control-directed backtracking technique that goes back over uncontrollable paths of arbitrary but finite length, from bad states or states that close bad cycles, to prune the search space more efficiently. Most of the states on these paths are not expanded further.[1]

- Incremental creation of the state space of the controller and calculation of the feedback function by determining incrementally states at which controllable events must be enabled (this is dual to determining those that must be disabled).

The obtained controller is represented by a pair $(\mathcal{M}, \phi)$, where $\mathcal{M} = (Q, A, \delta, q_0)$ is a transition structure and $\phi : Q \rightarrow \Gamma$ a feedback function determining enabled actions. The combination of a process and a controller constitutes a closed-loop system. In fact, $\mathcal{M}$ mimics the behavior of the concurrent execution of processes while ϕ determines the set of permissible process actions for each step of the execution of the closed-loop system. The reader is referred to [1] and [2] for more details. Computation has been performed with our own software tool, called Temporal Controller Synthesis Tool (TCST), implemented in Lisp.

Figure 6 illustrates the transition structure synthesized for ADC. States are represented as circles embedding propositional symbols. Initially, the controller is in state 1. From this state, it senses the current antenna position and compares the result with the target. The outcome is uncontrollable and the controller may handle any of the three possible alternatives (high, good, or low).

Let us consider in more detail the case where the position is high. In state 2, the precondition of constraint C1 is satisfied. The transition structure of Figure 6 presents two choices of trajectories that may be selected according to the fact that either constraints C3 is taken into account or not by the synthesis algorithm. If C3 is taken into account, the precondition of C3 is true in state 2. Satisfaction postcondition of C3 is required in the next state. In other words, activation of the rotor in the down direction is forced. The controller moves to state 6. The antenna is eventually lowered to the good position, moving from state 6 to state 7. Because of constraint C2, the rotor is stopped, moving from state 7 to state 3.

The process remains in state 3 (action *Wait*) until a new sensing of the current antenna position is required (action *Deltpos*) which takes the controller back to the initial state.

[1]The control-directed backtracking technique is the main difference between a similar algorithm described in [3], where uncontrollable transitions are abstracted over by nondeterministic transitions.

Now, let us consider a scenario in which constraint C3 is not interpreted by the synthesis algorithm. It is perfectly valid, although not optimal, for the controller to behave as follows from state 2. Start the rotor in the up direction for a while (transition from state 2 to state 5). Stop the rotor (transition from state 5 to state 10). Then, finally, start the rotor in the down direction (transition from state 10 to state 6).

The point to stress is here is that in the MTL formula modeling the control requirements, there may be some sub formulas that may not be required to insure synthesis of a valid solution. They are, however, helpful because they represent heuristics guiding the algorithm to synthesize more quickly a less complex solution. Indeed, in the current example, solely constraints C1 is mandatory for the validity of a solution. Although, conjunction of C1 with C2 to C7 contributes to limit the number of generated states and compute a more optimal solution.

In Figure 6, dotted arrows represent additional trajectories generated if only C1 is considered (the dashed arrow must not be considered in this solution). A total of ten states is generated. Conjunction with C2 to C7 leads to a controller without the dotted transitions and states 5 and 10 (but with the dashed transition). Hence, the controller has only 8 states.

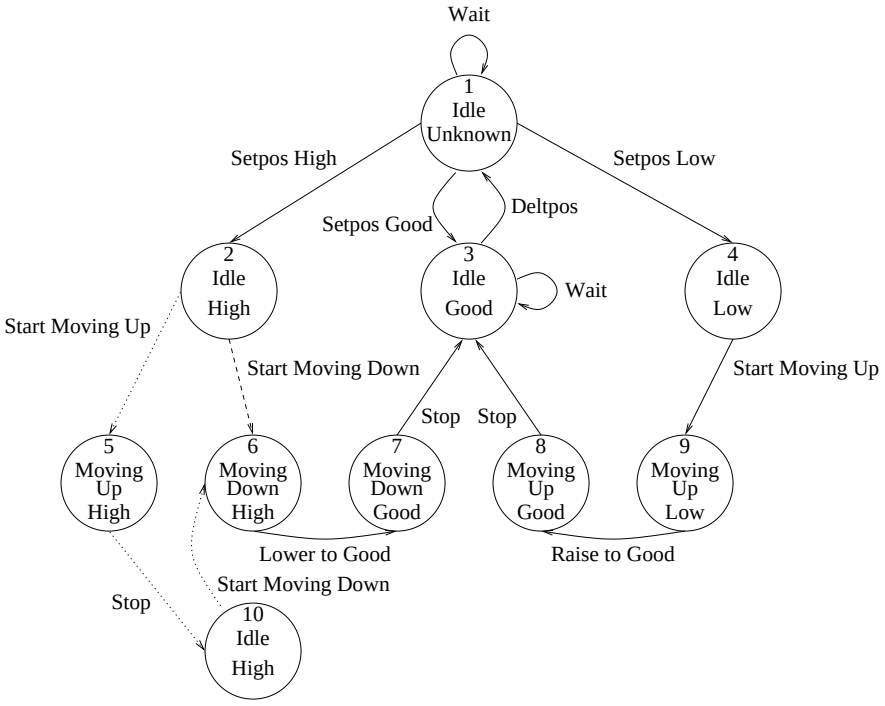

Figure 6: Controllers

The feedback function ϕ can be easily computed from the transition structure of the controller. Actions that are permissible in a given state are those labeling transition outgoing from that state. Note that if, in a given state, there are active uncontrollable actions, only them are permitted in that state. Otherwise, only one controllable action

is permitted. In this experiment, TCST did not compute a maximal controller in the traditional control theoretical sense (i.e., permit as many actions as possible). The controllable actions may be optimal though in a practical sense: e.g., enabling actions with the lowest cost for instance. This is in the line of AI or decision optimization concerns, as discussed in [3]. Nevertheless, it is possible to run TCST with an optional parameter requiring it to compute a maximal solution whenever there exists one. As in [6], such a maximal solution does not always exists when liveness constraints are involved.

4 Conclusion

We presented a simple realistic synthesis example. Calculation of the controller has been performed with the TCST tool within a period of 0.13 seconds. This suggests that the synthesis approach is feasible. In addition, formalization of our problem before implementation helped us to uncover errors in our design.

Future work is required to improve the realism of our model. For example faults are not taken into account.

References

[1] M. Barbeau, F. Kabanza, and R. St-Denis, "A method for the synthesis of controllers to handle safety, liveness, and real-time constraints," Technical report number 196, Département de mathématiques et d'informatique, Université de Sherbrooke, 1997.

[2] M. Barbeau, F. Kabanza, and R. St-Denis, "Supervisory control synthesis from metric temporal logic specifications," *Proc. Thirty-third Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, September 1995, pp. 96–105.

[3] F. Kabanza, M. Barbeau, and R. St-Denis, "Planning control rules for reactive agents," *Artificial Intelligence*, vol. 95, no. 1, August 1997, pp. 67–113.

[4] R. Koymans, "Specifying real-time properties with metric temporal logic," *Real-time Systems*, vol. 2, no. 4, August 1990, pp. 225–299.

[5] B. Selic, G. Gullekson, and P. T. Ward, *Real-time Object-oriented Modeling*, John Wiley & Sons, Inc., 1994.

[6] J. G. Thistle and W. M. Wonham, "Control of infinite behavior of finite automata," *SIAM J. Control and Optimization*, vol. 32, no. 4, pp. 1098–1113, 1994.