Comparing an Evolved Finite State Controller for Hybrid System to a Lookahead Design

Jean-François Dupuis and Zhun Fan

Abstract— **This paper presents a comparison of an evolutionary methodology for evolving finite state controller to the lookahead controller for hybrid system. To illustrate the advantages and disadvantages of both controllers two case studies, namely a two-tanks system and a single-input double-output DC-DC converter circuit, are used for comparison.**

I. Introduction

Mechatronic systems are the complete integration of mechanics, electronics and information processing. Tight integration of these domains make them highly dependent on each other. Design choices in one domain affect the performance of the other domains. Therefore, design of such a system usually requires iterations in each domain [1] in order to find an optimal balance between the basic mechanical structure, sensor and actuator implementations, automatic digital information processing and overall control. In an effort to automate the generation of mechatronic systems spanning multiple domains, the use of the bond graph [2] representation and genetic programming for search was proposed [3]–[8]. In this previous work, interesting results on a variety of case studies were presented. However, in all of these examples of automated design, the information processing capability was quite limited. In fact, all the case studies presented were of time-driven systems. The addition of an event-driven controller to a mechatronic system results in a more intelligent device. Mixing both time-driven and event-driven control in a hybrid controller increases the design complexity achievable in comparison to current automated design systems.

Such hybrid systems may be viewed as an extension of a classical time-driven system, typically modelled through differential or difference equations, with occasional discrete events causing a change in its dynamic behaviour. When such an event takes place, the system is thought of as switching from one operating mode to another. Hybrid or switched bond graph representations [9], [10] have been proposed to model physical dynamic systems with discontinuity. This hybrid system representation extends the normal bond graph by adding a switch element that can act as a flow source or as an effort source according to the current system state or controller action. A supervisory control system [11], [12] can thus extend the decision making capability of a normal bond graph in a compact way. The resulting hybrid system can be represented as in Figure 1 [12], where the hybrid bond graph acts as the plant and the controller manages the state

Both authors are with the Technical University of Denmark, Nils Koppels Alle, Building 426, 2800 Kgs. Lyngby, Denmark jedu@man.dtu.dk, zhfa@man.dtu.dk

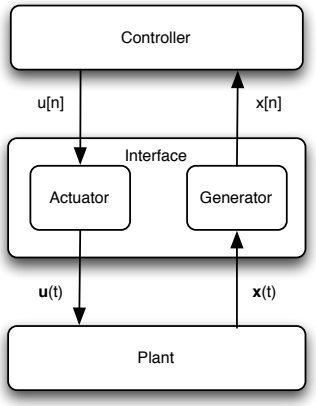


Fig. 1. Hybrid control system.

of the switch element in the bond graph. The interface acts as a translator between the continuous space and the discrete controller space, whereas the switches in the hybrid bond graph act as the actuators. The generator is usually part of the controller design as it needs to generate meaningful symbols according to the state of the plant.

Several techniques have been proposed over the years to automate the design of controllers. Neural networks with fixed and open topologies [13] are often seen for time-driven systems. However, for supervisory controllers, finite state automata (FSA) are usually better at representing the logical relationships in the system. The interest in evolution of FSA is a very old one, having started almost 50 years ago with the work of Larry Fogel [14], but is still active today [15]–[17].

This article compares the performance and domain of application of an evolved FSA controller [18] with a lookahead controller [19], [20], in order to determine the best approach to the controller synthesis problem that arises in the development of a framework for the automated design of hybrid system. A two-tank system and a single-input double-output DC-DC converter circuit are used as study cases to illustrate the advantages and disadvantages of the two approaches. The remainder of the paper is organized as follows the next section describes the two tanks systems and the performance achieved with the different controller. Then Section III presents the single-input double-output DC-DC converter case studies and the relevant results. Finally, Section IV concludes the paper with the lessons learned from this comparison and provides some perspectives and future work.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

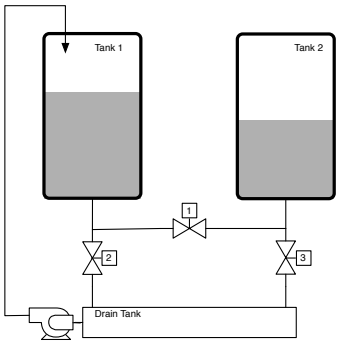


Fig. 2. The two-tank system.

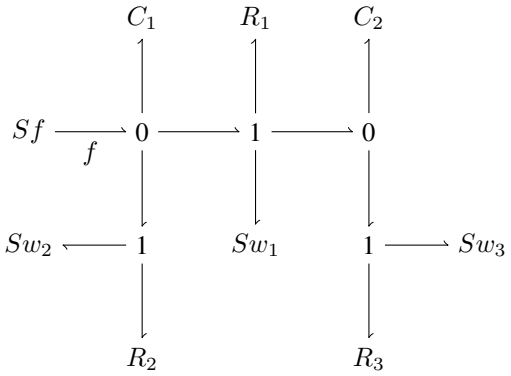


Fig. 3. The hybrid bond graph of the two-tank system.

II. Two-Tank System

Multiple-tank systems are often encountered in the research literature concerning non-linear multi-variable feedback control, as well as in fault diagnosis literature [21], [22]. The mechanical simplicity and the ease of getting physical insight into the system behaviour, combined with the achievable control complexity, make the multi-tank problem a very attractive testbed. Therefore, a two-tank system was defined to test the controller synthesis methodology presented in this article. Figure 2 shows the actual configuration. A pump is continuously filling the first tank at a constant flow rate, and a set of valves allows draining each tank independently and also allows bidirectional transfer from one tank to the other.

The hybrid bond graph of this two-tank system is shown in Figure 3. The pipe and valve restrictions are represented by the resistive components, R, while the tanks are represented by the capacitances, C. The valves are simply modelled by switch components, Sw. These switches act as a 0-flow source or a 0-effort source, depending on their state. A 0-flow source imposes a flow equal to zero at the junction connected to it, therefore the valve is said to be closed as no fluid is able to pass through it. A 0-effort source indicates that the switch does not impose any restriction on the flow. The valve is then said to be open.

A. Evolved FSA controller

The controller trained for this two-tank system is a FSA having a fixed number of five states as established by an invariant analysis [23]. Each state corresponds to a possible switch configuration in the hybrid bond graph.

The input symbols are generated by the interface when the state variables cross predefined surfaces in the state space. These surfaces are separated into two sets, one for each tank, and are defined as follow :

$$\begin{aligned} h_{i1} &= y_i - T_i + \delta \\ h_{i2} &= y_i - T_i - \delta \end{aligned} \tag{1}$$

in which T_i is the desired level for tank i and δ is the tolerance about the target. Each set separates the space into three regions depending on whether the level of the tank i is above, below or at its target. Then nine symbols are formed from the logical conjunction of the two sets.

The transitions used in this implementation of the FSA do not have any actions associated with them; they simply specify what the new state will be, in reaction to the input symbol received. Therefore, the controller can be expressed as a simple matrix with the states as row indices and the input symbols as column indices. The elements of the matrix then specify the next state to which the FSA should move. An initial state need also to be defined outside this matrix to complete the definition.

1) Fitness evaluation: When evaluating the fitness of the evolved controllers, for each simulation case, the system is integrated for a period of 15 seconds. An objective function $\phi(i)$ is computed for each tank at the end of the simulation, based on the level errors :

$$\phi(i) = \int (y_i - T_i)^2 dt \tag{2}$$

Later, the fitness of this simulation case is defined as the fitness of the tank with the worst error:

$$\Phi = \frac{1}{\max(\phi(i))} \tag{3}$$

Looking at the tank with the worst error proved to be a more successful approach than looking at the average fitness of the two tanks. When looking at the average, the incentives to reach the target were not strong enough, and the evolution was ceasing after finding compromises between the errors of the two tanks. Most of the time it was observed that one of the tanks sacrificed itself in order to get excellent performance by the other. The really poor performance of the sacrificed tank was then compensated for by the performance of the other tank, yielding a high average. However, looking at the worst tank disallows such behaviour and enforces a good performance of both tanks.

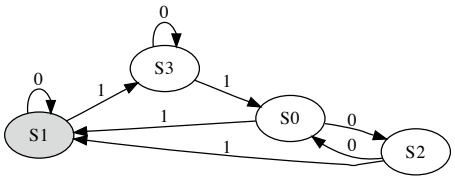
2) Multiple simulation cases: In order to obtain controllers that generalize to cases outside the training set, the controller is tested on several simulation cases. For each case the fitness is computed as defined in the previous section and again, the worst case is used as the fitness processed by the selection operator.

	0	1
0	2	1
1	1	3
2	0	1
3	3	0

(a) Transition table.

10 01 01 11 00 01 11 00 01

(b) Genotype



(c) Phenotype

Fig. 4. Encoding example

3) Genetic algorithm implementation: The evolution of the controller uses a simple genetic algorithm (GA) with standard one-point crossover, bit-flipping mutation and tournament selection. The transition table matrix described in section II-A is encoded in a bit string with three bits per element, with three extra bits at the end to define the initial state. The evolved bit string is thus 219 bits long. Figure 4 shows the proposed encoding approach used in a simpler case with only four states and two input symbols. In this example, only two bits per element are used. One can see that the elements of the transition matrix are written row by row to the associated genotype.

The implementation of this GA experiment was done using the Open BEAGLE C++ framework [24] and the evaluation of the fitness was distributed on a cluster of computers using MPI [25].

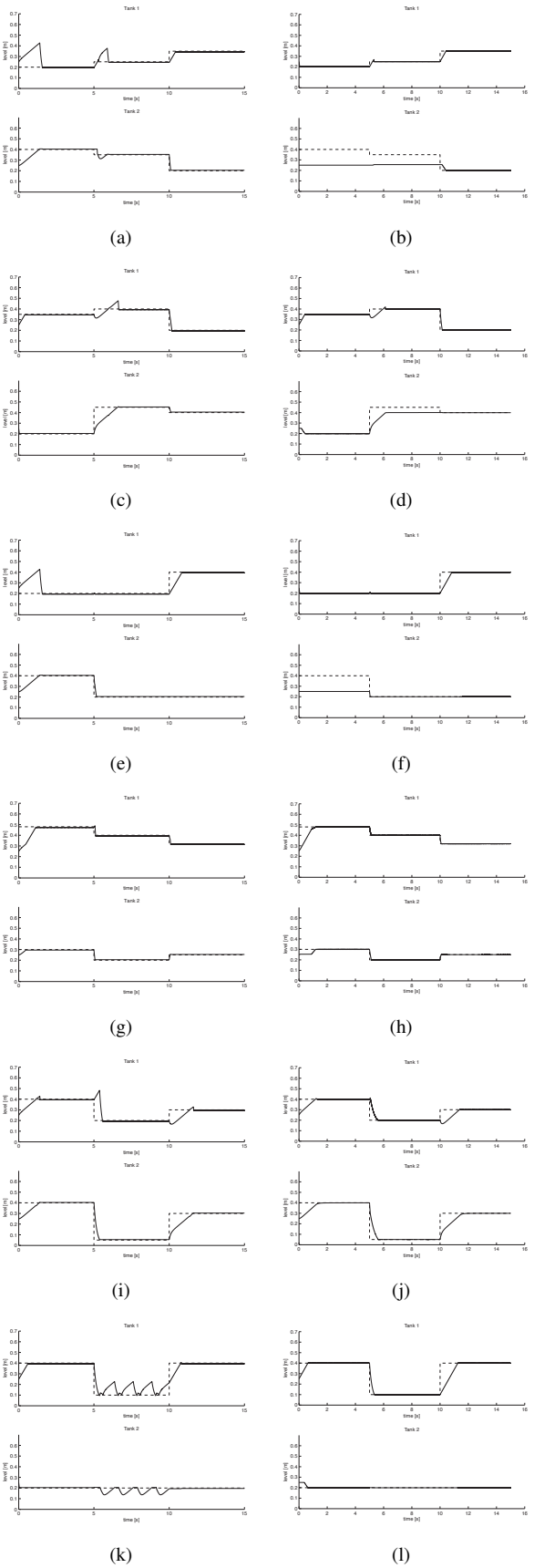
B. The lookahead controller

A single step lookahead controller was also implemented to control the two-tanks system. This controller will at run time, for each time step, look at the system responds for each switch configuration and then choose the best state in which the system should go to reach the target. The best state is defined as the one whose vector field points closer to the set point x_d. Thereby, the control action minimizes the cosine of the angle between the current target direction $x - x_d$ and the system trajectory $x - x_i$ for each state i.

C. Results

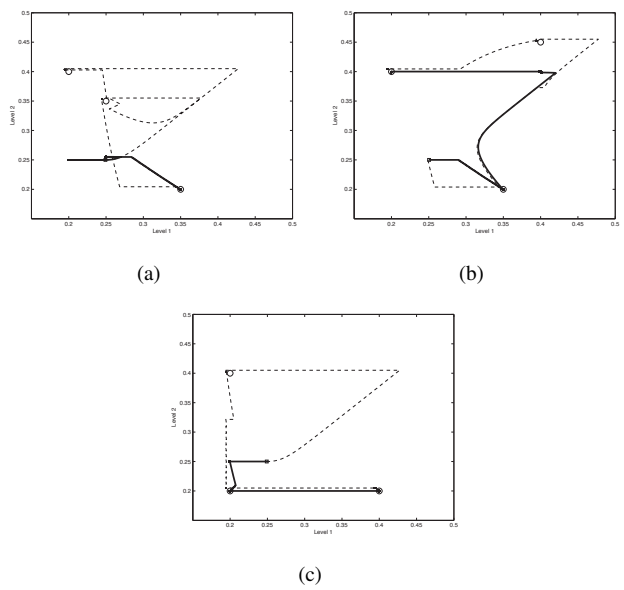
For the experiments reported, the two-tanks controller was asked to keep the fluid levels of the tanks within their target regions. The system responds to different cases are shown in Figure 5.

The performance of the evolved controller is quite satisfactory as the set points are successfully reached in a minimal amount of time. There are still some imperfections that could be corrected, such as the ripples seen in 5(k) and the overshoot



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 5. Controller behaviour on the target set. On the left, the best evolved FSA controller. On the right, the one step lookahead controller. The tanks levels and desired targets are represented by a solid and dashed line respectively.



(a) (b) (c)

Fig. 6. State-space trajectory for the two-tanks system exhibiting failure of the lookahead controller. The FSA controller and the lookahead controller are represented by a dashed line and a solid line respectively. The targets are marked by circles and the initial state is marked by a square.

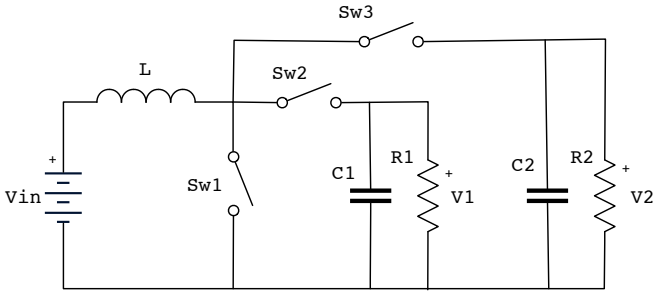


Fig. 7. Single-input double-output DC-DC converter circuit.

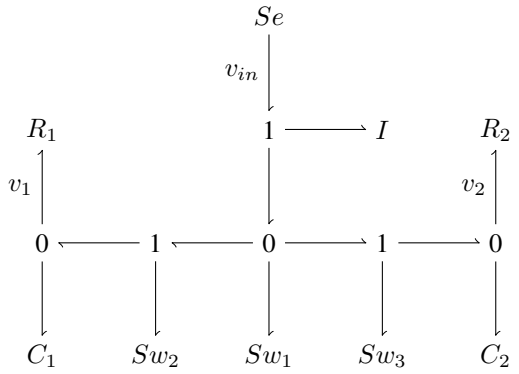


Fig. 8. The hybrid bond graph of the double output DC-DC converter circuit.

on the way down of the first tank as seen in Figures 5(a) and 5(i). However, the lookahead controller fails completely at reaching its targets on the second tank in the first three cases as seen in Figures 5(b), 5(d) and 5(f), where their state space trajectory is illustrated in Figure 6.

In these cases, the desired target asks for a level in tank 2 higher that in tank 1. From the inspection of Figure 2, the only way to raise the level in the second tank is to first raise the level in tank 1 and then transfer the fluid between them. Consequently, the level in tank 1 needs to go away from its target in order to help to reach the objective of the other tank. Therefore, the lookahead controller can't choose the right state as it's not the one going in the direction of the target. The controller would require a much longer lookahead in order to establish the correct sequence of action. But when dealing with a system exhibiting a small time constant, the amount of simulation time required to go through a deep tree of possible state can easily be too long.

On the other hand, even though this is not rewarded by the fitness function, as tank 1 is accumulating much error during this process, the evolution was able to find the correct control sequence that will meet the target as this is the best compromise for achieving the best fitness.

III. DC-DC Converter

Switching circuits are often employed in power applications as they represent a very effective way to transform energy. The single-input double-output DC-DC converter [19] shown in Figure 7 is an example of such a circuit. The hybrid bond graph equivalent, used as a simulation tool, is also shown in Figure 8. The purpose of the circuit is to supply the loads R_1 and R_2 with voltage V_1 and V_2 that are both higher than the source voltage V_{in}. This is done by storing energy in the inductor L when Sw_1 is closed while Sw_2 and Sw_3 are open. Then, the stored energy is transferred to the capacitors C_1 or C_2 by closing Sw_2 or Sw_3, while Sw_1 is kept open. In order to provide sufficient energy to the load resistances, the switching must occur continually with controlled timing.

The circuit parameters are the same as the one used by Senesky [19], that is $L = 75\mu\text{H}$, $R_1 = 6.25\Omega$, $R_2 = 34.1\Omega$, $C_1 = 800\mu\text{F}$ and $V_{in} = 1.5\text{V}$. The desired output voltages are $V_{1d} = 1.875\text{V}$ and $V_{2d} = 3.75\text{V}$. The current range is required to be $[0, 2.5]\text{A}$.

A. Evolved FSA controller

The controller trained for the DC-DC converter is also a FSA having a fixed number of three states as Sw_2 and Sw_3 are not closed at the same time.

The input symbols are generated by the interface when the state variables are crossing predefined surfaces. Those surfaces are defined in the same way as described for the two-tanks systems in section II-A, that is, a tolerance is established for all target values and an input symbol is generated each time the state variables cross the tolerance limits. In this case, the tolerances on the output voltages were set to 0.01V. In the case of the current constraint, an input symbol is generated each time the current through the inductor goes out of the specified range of $[0, 2.5]\text{A}$. As a result, there are nine surfaces creating 27 regions in the state space.

1) Fitness evaluation: When evaluating the fitness of the evolved controllers, the system is integrated for a period of

50ms. An objective function $\phi(k)$ is computed for each voltage output at the end of the simulation, based on their errors :

$$\phi(k) = \int (V_k - V_{kd})^2 dt : k \in 1, 2 \tag{4}$$

Also, a third objective function $\phi(3)$ is computed based on the current i through the inductor as follow:

$$\begin{aligned} f(i) &= \begin{cases} i - 2.5 & : i > 2.5 \\ i & : i < 0 \\ 0 & : otherwise \end{cases} \\ \phi(3) &= \int f(i)^2 dt \end{aligned} \tag{5}$$

Later, the fitness for this controller is defined as the worst objective function:

$$\Phi = \frac{1}{\max(\phi(k))} : k \in 1, 2, 3 \tag{6}$$

Again, looking at the worst case proved to be the best approach to push the search algorithm to reach the desired target.

2) Genetic algorithm implementation: The evolution is setup in the exact same way as the two-tanks system, as described in section II-A.3. However, this time the evolved bit string is 246 bits long.

B. The lookahead controller

Again, a single step lookahead controller was implemented for the DC-DC converter. The controller is defined in the exact same way as the two-tanks systems, as described in section II-B.

C. Results

The state variables evolution under the control of the evolved FSA and the lookahead controller for the single-input double-output DC-DC converter are shown in Figure 9.

As we can see, the system shows much more ripples using an evolved FSA controller than using a lookahead one. This can mostly be due to the fact that the FSA is limited by the set of input symbols defined by the designer. Therefore, the controller can only react when a defined surface in the state space is crossed. So, if these surfaces are suboptimal or even wrong at representing the meaningful dynamics for the control task, the controllability of the system is jeopardized. Therefore, the division of the state space is a crucial step in the design of a FSA based controller. It's not surprising that previous work attempts to develop some methodology to divide the state space in an optimal way by looking at the control invariance of the system [12].

On the other hand, the lookahead controller is able to minimize the ripples on the output in an optimal way. It's able to achieve such performance by having direct access to the state variables to make the next control action. However, the design of the lookahead controller requires a steady state analysis of the circuit to establish the current target. Without such analysis, the desired target can be unreachable and therefore makes the lookahead controller fail to control the system.

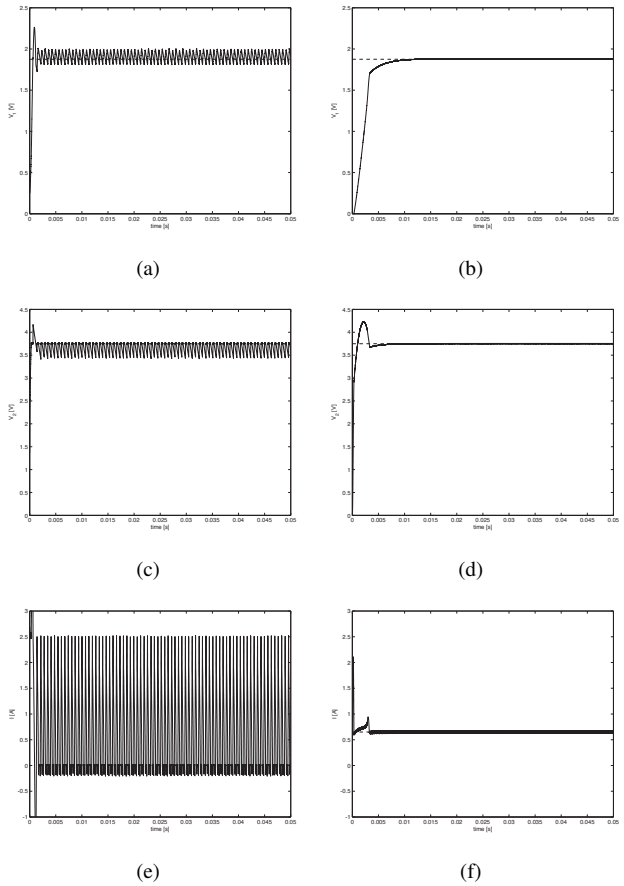


Fig. 9. State trajectories of the single-input double-output DC-DC converter. On the left, the best evolved FSA controller. On the right, the one step lookahead controller.

IV. Conclusion

As we can see from the experiments with the two study cases presented, neither of the two controller strategies work perfectly in both cases. However, each of them represent a successful approach to control a certain type of system. For instance, the FSA based controller provides a better approach when the system to control presents a vector field that is not flowing in a direction close to the target. In this case, the controller could make an efficient use of prior knowledge about the vector field of the system. On the other hand, if the system presents a favourable vector field, the lookahead controller can exploit the knowledge on the future evolution of the state variables at every time step. The controller doesn't need to wait before the system reaches a predefined frontier in the state space to react. This type of controller can be much more responsive than the FSA based one.

The weakness of the two approaches should also be considered in further research. For instance, the input symbols of the FSA generated by the interface should be better defined. They are still too intuitively defined by the designer. They would be better defined if a more rigorous approach was used. However, the automated techniques for analysing the vector field of a hybrid system in order to establish the better

transition time still need some development. For instance, in [26] a method to partition the state space to create a state-feedback FSA controller is proposed, but the computational time required for a single target is too important. On the other hand, the usability of a lookahead controller is limited by the amount of simulation needed at runtime to establish the correct control decision. This gets more restrictive when dealing with systems presenting a small time constant. The accuracy of the lookahead prediction can also be increased by allowing a deeper simulation tree. However, the simulation time required grow exponentially with the lookahead time. Hence, the domain of application of a lookahead controller is confined to system having slow dynamics.

With the objective in mind of generating hybrid bond graph design within an evolutionary framework, there is a need for a method to automatically generate switch controllers that can be used to correctly evaluate the generated bond graph design. One approach could be to exploit the fact that the lookahead controller doesn't need to be tuned to work well in most situation. Therefore, the lookahead controller can rapidly be used to assess the performance of a new hybrid system design without wasting too much time on tuning a controller for a poor design. Also, the design of the hybrid system could take into account the limitations of the controller by optimizing the vector field such that the limitations of the lookahead controller are avoided. Another approach, could be to develop an evolutionary system that would include both the FSA based controller and the simple lookahead one in a single controller population that would be co-evolved with the hybrid bond graph population.

REFERENCES

[1] Q. Li, W. J. Zhang, and L. Chen, "Design for control-a concurrent engineering approach for mechatronic systems design," *Mechatronics, IEEE/ASME Transactions on*, vol. 6, pp. 161–169, 2001.

[2] D. Karnopp, D. Margolis, and R. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*, 4th ed. Hoboken, New Jersey: John Wiley and Sons, 2005.

[3] K. Seo, Z. Fan, J. Hu, E. D. Goodman, and R. C. Rosenberg, "Toward an automated design method for multi-domain dynamic systems using bond graphs and genetic programming," *Mechatronics*, vol. 13, no. 8-9, pp. 851–885, 2003.

[4] Z. Fan, J. Wang, and E. Goodman, "Exploring open-ended design space of mechatronic systems," *International Journal of Advanced Robotic Systems*, vol. 1, pp. 295–302, 2004.

[5] J. Wang, Z. Fan, J. P. Terpenny, and E. D. Goodman, "Knowledge interaction with genetic programming in mechatronic systems design using bond graphs," *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, vol. 35, pp. 172–182, 2005.

[6] J. Wang, Z. Fan, J. P. Terpenny, and E. D. Goodman, "Cooperative body-brain coevolutionary synthesis of mechatronic systems." *AI EDAM*, vol. 22, no. 3, pp. 219–234, 2008.

[7] Z. Fan, K. Seo, J. Hu, E. D. Goodman, and R. C. Rosenberg, "A novel evolutionary engineering design approach for mixed-domain systems," *Engineering Optimization*, vol. 36, no. 2, pp. 127 – 147, 2004.

[8] Z. Fan, J. Wang, S. Achiche, E. Goodman, and R. Rosenberg, "Structured synthesis of mems using evolutionary approaches," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 579–589, 2008.

[9] P. J. Mosterman, "Hybrid dynamic systems: A hybrid bond graph modeling paradigm and its application in diagnosis," Ph.D. dissertation, Vanderbilt University, 1997.

[10] J.-E. Strömberg, S. Nadjm-Tehrani, and J. Top, "Switched bond graphs as front-end to formal verification of hybrid systems," *Hybrid Systems III*, pp. 282–293, 1996.

[11] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987. [Online]. Available: http://link.aip.org/link/?SJC/25/206/1

[12] X. Koutsoukos, P. Antsaklis, J. Stiver, and M. Lemmon, "Supervisory control of hybrid systems," in *Proceedings of the IEEE*, vol. 88, 2000, pp. 1026–1049.

[13] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/106365602320169811

[14] L. J. Fogel, "Autonomous automata," *Industrial Research*, vol. 4, pp. 14–19, 1962.

[15] D. Ashlock, *Evolutionary Computation for Modeling and Optimization*. Springer New York, 2006, ch. Evolving Finite State Automata, pp. 143–166.

[16] K. Benson, "Evolving finite state machines with embedded genetic programming for automatic target detection," *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 2, pp. 1543–1549 vol.2, 2000.

[17] B. D. Dunay, F. E. Petry, and B. P. Buckles, "Regular language induction with genetic programming," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 1994, pp. 396–400 vol.1.

[18] J.-F. Dupuis, Z. Fan, and E. Goodman, "Evolved finite state controller for hybrid system," in *GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2009, pp. 105–112.

[19] M. Senesky, G. Eirea, and T. J. Koo, "Hybrid modelling and control of power electronics," *Hybrid Systems: Computation and Control. 6th International Workshop, HSCC 2003. Proceedings (Lecture Notes in Computer Science Vol.2623)*, pp. 450–465, 2003.

[20] S.-L. Chung, S. Lafortune, and F. Lin, "Limited lookahead policies in supervisory control of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1921–1935, 1992.

[21] J. Wu, G. Biswas, S. Abdelwahed, and E. Manders, "A hybrid control system design and implementation for a three tank testbed," *IEEE Conference on Control Applications*, pp. 645–650, 2005.

[22] M. Sainz, J. Armengol, and J. Vehi, "Fault detection and isolation of the three-tank system using the modal interval analysis," *Journal of Process Control*, vol. 12, no. 2, pp. 325–338, 2002.

[23] J.-F. Dupuis and Z. Fan, "Evolved finite state controller for hybrid system in reduced search space," *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 833–838, 2009.

[24] C. Gagné and M. Parizeau, "Genericity in evolutionary computation software tools: principles and case study," *International Journal on Artificial Intelligence Tools*, in-press 2006.

[25] *MPI: A Message-Passing Interface Standard*, Message Passing Interface Forum, http://www.mpi-forum.org, June 2008.

[26] J. Stiver, X. Koutsoukos, and P. Antsaklis, "An invariant based approach to the design of hybrid control systems," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 5, pp. 453–478, 2001.