XSpeed: A Tool for Parallel State Space Exploration of Continuous Systems on Multi-core Processors

Amit Gurung*

Supervisor: Rajarshi Ray
*(Department of Computer Science & Engineering
National Institute of Technology Meghalaya, India)
Email :*amitgurung@nitm.ac.in*

Joint work with

Rajarshi Ray*, Binayak Das*, Ezio Bartocci**, Sergiy Bogomolov***
and
Radu Grosu**
**(Vienna University of Technology, Austria)
***(Institute of Science and Technology, Austria)

Mysore Park Workshop, Feb 1-4, 2016

Overview

1. Parallel Reachability Analysis of Continuous Systems
   - Motivation of Reachability Analysis
   - Computation of Reachable States
   - Set Representation
   - Support Functions
   - Reachability Analysis using Support Functions
   - Our Contribution
2. Experimental Results
3. Conclusion and Ongoing Work

Motivation of Reachability Analysis

- An approach of analysing the behaviour(**safety**) of systems based on the computation of all **reachable states**.
- **Our Focus**: Continuous linear systems where the dynamics is of the form $\dot{x} = Ax(t) + u(t), u(t) \in U, x(0) \in X_0$

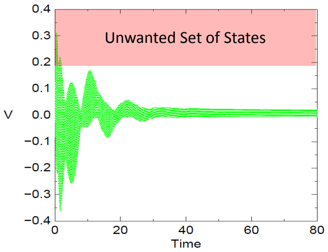
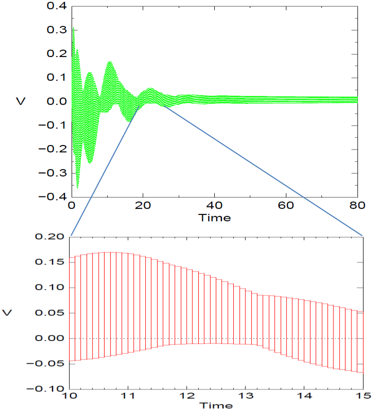


Figure: Reachable state-space of a 28 dimensional Helicopter Controller System generated using the tool SpaceEx.

- For e.g. time horizon of 1000 units divided into 10e4 intervals using octagonal approximation, SpaceEx does not complete the reachable set computation even in 1 hour on a modern pc

Computation of Reachable States



- Divide the time horizon into N intervals.
- Compute all possible reachable states until a fixed point is found or within a specified time bound.
- Each reachable state is represented by some efficient set representation.

Set Representation

Some of the methods used in the literature are

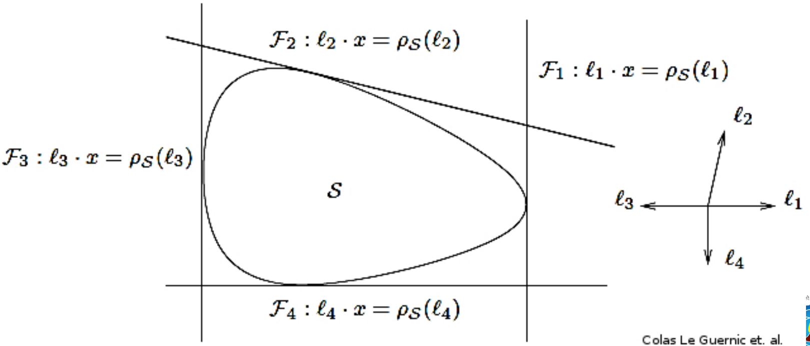
- Support Functions
- Zonotopes
- Ellipsoids
- Polytope

Support Function algorithm shows promising scalability and precision for convex sets representation.

Support Functions as Set Representation

- The support function of any compact convex set $S \subseteq \mathbb{R}^d$, denoted by $\rho_S : \mathbb{R}^d \to \mathbb{R}$ defined as

$$\rho_S(\ell) = max_{x \in S}(\ell \cdot x) \quad (1)$$



Colas Le Guernic et. al.

Support Functions as Set Representation

- Any compact convex set S can be represented by its support function.

$$S = \bigcap_{\ell \in R^n} (\ell \cdot x) \leq \rho_S(\ell) \tag{2}$$

- Practically, an approximation of a convex set is obtained by limiting the number of possible directions from $\ell \in \mathbb{R}^n$ to some finite set, say $\mathcal{D}$.

$$S = \bigcap_{\ell \in \mathcal{D}} (\ell \cdot x) \leq \rho_S(\ell) \tag{3}$$

Support Functions as Set Representation

- The choice of the number of directions $\mathcal{D}$ enables us to decide the desired precision (over efficiency).

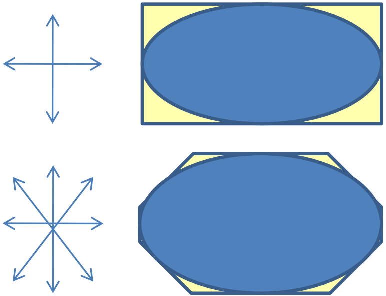
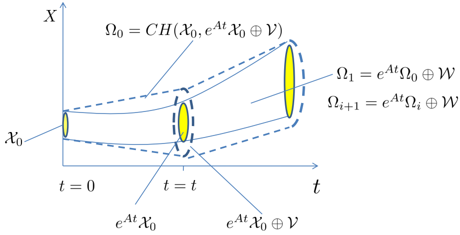
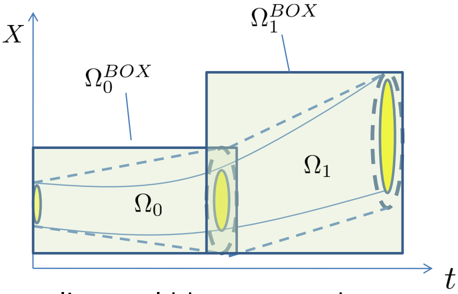


Figure 1): With $\mathcal{D}$ as bounding-box directions
Figure 2): With $\mathcal{D}$ as octagonal directions

Reachability Analysis using Support Functions

- Polyhedral approximations are computed from the support function representation using finite samplings in bounding directions.





Observation: Each support function sampling could be computed independent of each other.

Our Contribution

Parallel Computation of Reachable Set

1. Function samplings over template directions in parallel

Our Contribution

Parallel Computation of Reachable Set

1. Function samplings over template directions in parallel
2. Parallel State-Space Exploration in Sliced Time Horizon

Our Contribution

Parallel Computation of Reachable Set

1. Function samplings over template directions in parallel
2. Parallel State-Space Exploration in Sliced Time Horizon
3. Lazy evaluation of support functions in GPU

1) Function samplings over template directions in parallel

1. X_0 and U are polytopes.
2. Parallel samplings boil down to parallel LP solving with the constraint space but different objective functions.
3. We use the open source GLPK library as an LP Solver but **GLPK is not thread safe**.
4. We identify the shared data in the implementation and modified to *thread local*.

2) Parallel State-Space Exploration in Sliced Time Horizon

1. Time horizon T is sliced into equal intervals of size $T_p = T/N$, N being the degree of parallelism.
2. We compute reachable states for each of these intervals in parallel.
3. The initial set of states for each reachability computation is computed a priori.

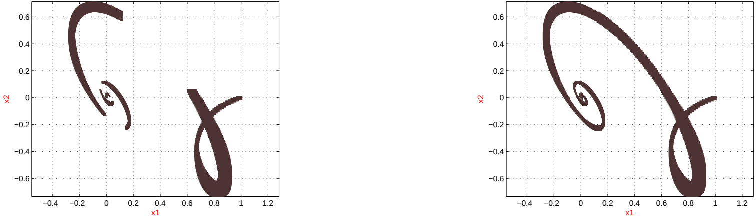
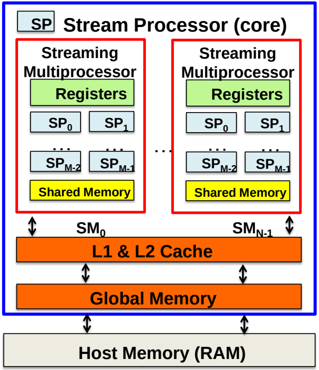


Figure: shows that a time horizon of 5 units is sliced into five intervals each of size 1 unit. a) Reachable states computed by individual threads in 0.75 time unit. b) computed by threads in 1 time unit.

3) Lazy evaluation of support functions in GPU

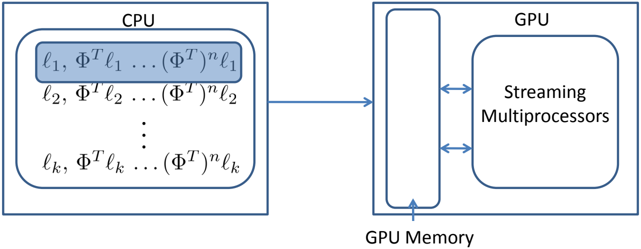
1. *Delaying support functions evaluation until all the directional arguments are computed.*
2. The support functions are then evaluated in parallel in GPU.
3. CUDA implementation.

3) Introduction to GPU (Graphic Processing Unit)



1. Large number of co-processors for graphics processing.
2. Recently used to accelerate scientific applications.
3. Works as SIMT(Single Instruction Multiple Threads) execution model.
4. E.g. Nvidia GeForce GTX 670 has 7 SMs each with 192 cores = 1344 cores in all.

3) Support Function computation in GPU



1. Compute all template directions along with their transformed directions a priori.
2. Offload these directions in batches into GPU's memory (based on the device capacity).
3. Compute support functions of these directions in GPU cores and transfer the results back to CPU.

3) Support Function computation in GPU

- We observed that designing an LP Solver in GPU for our benchmark models were not helpful.
- Computing support functions of a Hyperbox is efficient.
- We designed GPU *Kernel* to compute support functions of a Hyperbox.

Results ...

Results: Sampling Support Functions on Multicore processors

Models	Nos of Directions (Threads)	4 Core (8 threads with hyperthreading)			6 Core (12 threads with hyperthreading)		
		Seq Time	Par Time	Seq Vs Par	Seq Time	Par Time	Seq Vs Par
5 Dim Model	Box (2x5=10)	0.1946	0.087	**2.24**	0.3360	0.1012	**3.32**
	Oct (2 x5^2=50)	0.9581	0.337	**2.84**	1.5322	0.4013	**3.82**
	500	9.0011	3.095	**2.91**	14.0371	3.2431	**4.33**
Helicopter Controller (28 dim)	Box (2x28=56)	2.2892	0.5138	**4.46**	3.3600	0.6083	**5.52**
	Oct (2 x28^2=1568)	63.8640	13.7986	**4.63**	93.8370	13.6688	**6.87**
	3000	121.5703	26.7662	**4.54**	178.9130	26.0147	**6.88**

- Experimental results are on a **5-Dimensional Model and a Helicopter Controller benchmarks**

Results: Sampling Support Functions on Multicore processors

Models	Nos of Directions (Threads)	4 Core (8 threads with hyperthreading)			6 Core (12 threads with hyperthreading)		
		Seq Time	Par Time	Seq Vs Par	Seq Time	Par Time	Seq Vs Par
5 Dim Model	Box (2x5=10)	0.1946	0.087	**2.24**	0.3360	0.1012	**3.32**
	Oct (2 $x5^2$=50)	0.9581	0.337	**2.84**	1.5322	0.4013	**3.82**
	500	9.0011	3.095	**2.91**	14.0371	3.2431	**4.33**
Helicopter Controller (28 dim)	Box (2x28=56)	2.2892	0.5138	**4.46**	3.3600	0.6083	**5.52**
	Oct (2 $x28^2$=1568)	63.8640	13.7986	**4.63**	93.8370	13.6688	**6.87**
	3000	121.5703	26.7662	**4.54**	178.9130	26.0147	**6.88**

- Experimental results are on a **5-Dimensional Model and a Helicopter Controller benchmarks**
- *We see a speed-up of maximum of* 22.75× *on a Helicopter Controller benchmark and* 5.84× *on a 5-Dimensional Model when compared to SpaceEx(LGG) on a 4Core (8 threads) processor.*

Results:Parallelizing Sliced Time Horizon on Multicore processors

Two fold gain ...
1) Gain on Precision

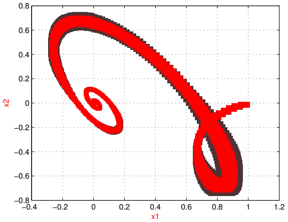


Figure: Five Dimensional System

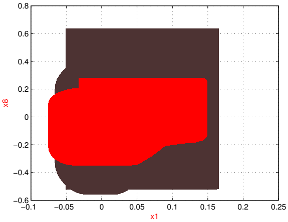


Figure: Helicopter Controller Model

Illustrating gain in precision with parallel state-space exploration (red-colour) over sequential (brown-colour) algorithm.

Results:Parallelizing Sliced Time Horizon on Multicore processors

2) Gain on Performance

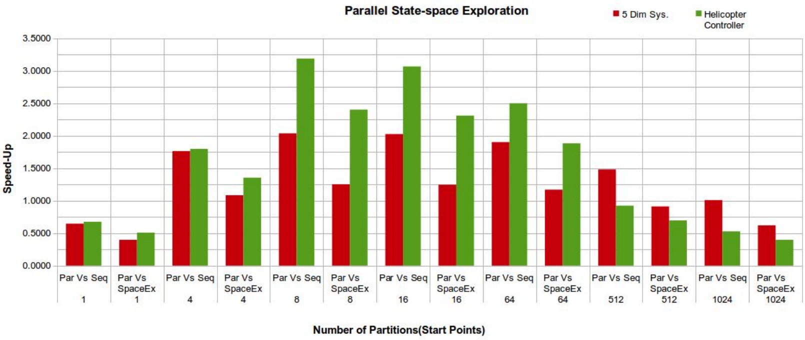


Figure: Illustrating gain in performance with parallel state-space exploration for different choice of slices

Results: Lazy evaluation of support functions in GPU

Models	Nos of Directions	Iterations	Time (in Seconds)			Speed Up	
			CPU-Sequential GLPK	**SpaceEx**	GPU &Multi-CPU Parallel	SpaceEx Vs GPU-Par	Seq Vs GPU-Par
5 Dimensional System	Box (2x5=10)	1000	0.1332	0.345001	0.0183	**18.82**	**7.27**
	Box (2x5=10)	2000	0.2870	0.686001	0.0287	**23.93**	**10.01**
	Oct (2 x5^2=50)	1000	0.7173	1.3990	0.0604	**23.15**	**11.87**
	Oct (2 x5^2=50)	2000	1.4620	2.8000	0.1189	**23.55**	**12.30**
	500	1000	6.6948	24.1711	0.5760	**41.96**	**11.62**
	500	2000	13.3296	39.5801	1.1144	**35.52**	**11.96**
	1000	1000	13.1282	59.9961	1.1210	**53.52**	**11.71**
	1000	2000	26.0223	94.2041	2.2196	**42.44**	**11.72**
Helicopter Controller (28 dim)	Box (2x28=56)	1000	1.4001	4.39901	0.1721	**25.56**	**8.14**
		1500	2.0778	7.26301	0.2495	**29.11**	**8.33**
		2000	2.7699	8.68501	0.3278	**26.50**	**8.45**
		2500	3.4440	11.0141	0.4053	**27.18**	**8.50**
	Oct (2 x28^2=1568)	1000	39.0893	123.794	4.2464	**29.15**	**9.21**
		1500	57.632	248.769	6.3213	**39.35**	**9.12**
	2000	1000	50.3673	187.825	5.3965	**34.80**	**9.33**
	3000	1000	75.0868	**311.652**	8.0545	**38.69**	**9.32**
		2000	149.3130	608.214	16.0922	**37.80**	**9.28**

Figure: Speed-up with lazy evaluation of support functions in GPU.

Conclusion and Ongoing Work

- We have implemented two variants of parallel implementation of support function algorithm in XSpeed.
- A lazy approach of evaluating support functions to achieve parallelism have also been implemented in GPU using the CUDA implementation.
- We have observed a significant performance gain in reachability analysis both in terms of speed-up and precision.
- Ongoing work is on Parallelization of Discrete jumps and Safety Verification of Hybrid Systems.

References

Rajarshi Ray and Amit Gurung
Parallel state space exploration of linear systems with inputs using XSpeed
In Girard, A., Sankaranarayanan, S.: 18th International Conference on Hybrid Systems: Computation and Control, HSCC'15.

Rajarshi Ray, Amit Gurung, Binayak Das, Ezio Bartocci, Sergiy Bogomolov and Radu Grosu
XSpeed: Accelerating Reachability Analysis on MultiCore Processors
In Hardware and Software: Verification and Testing- 11th International Haifa Verification Conference, HVC Nov-2015, Haifa, Israel.

Colas Le Guernic and Antoine Girard
Reachability Analysis of Hybrid Systems using Support Functions
In: Bouajjani, A., Maler, O. (eds.) CAV. Lecture Notes in Computer Science, Springer (2009)

Thank You