An Approximation Algorithm for Box Abstraction of Transition Systems on Real Vector Fields

(This is a revised version of a paper presented in SICE2009.)

Kunihiko HIRAISHI and
Koichi KOBAYASHI,

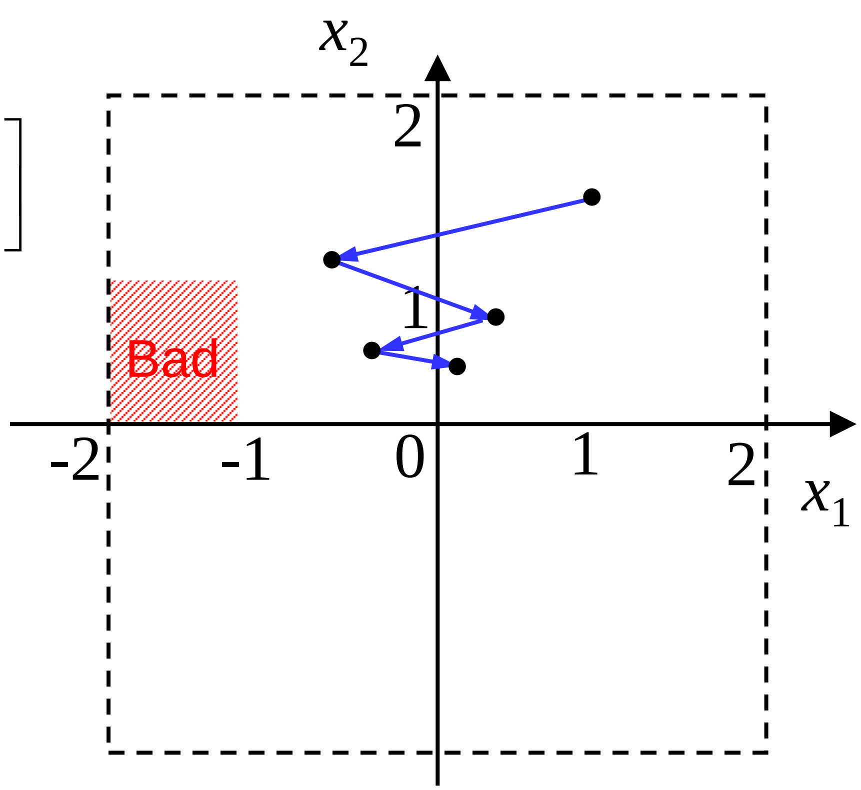
JAIST

Predicate abstraction for Hybrid Systems

- Predicate abstraction is a powerful technique for extracting finite-state models from infinite-state systems.
- Predicate abstraction has also been applied to the verification of hybrid systems [Alur00,Alur02,Alur06].
- Given a hybrid system with linear dynamics and a set of linear predicates, the verifier performs a search of *the finite discrete quotient* whose states correspond to the truth assignments to the input predicates.
- We propose a technique that can be used for *accelerating* the computation of abstract state spaces for hybrid systems.

Predicate abstraction: example (1)

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \alpha \cdot \begin{bmatrix} \cos\beta\pi & -\sin\beta\pi \\ \sin\beta\pi & \cos\beta\pi \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$



$\alpha = 0.8$,
$\beta = \ \ 1/6\ (x_1 \le 0,\ x_2 > 0)$;
$\beta = -1/6\ (x_1 > 0,\ x_2 > 0)$;
$\beta = \ \ 1/8\ (x_1 \le 0,\ x_2 \le 0)$;
$\beta = -1/8\ (x_1 > 0,\ x_2 \le 0)$.

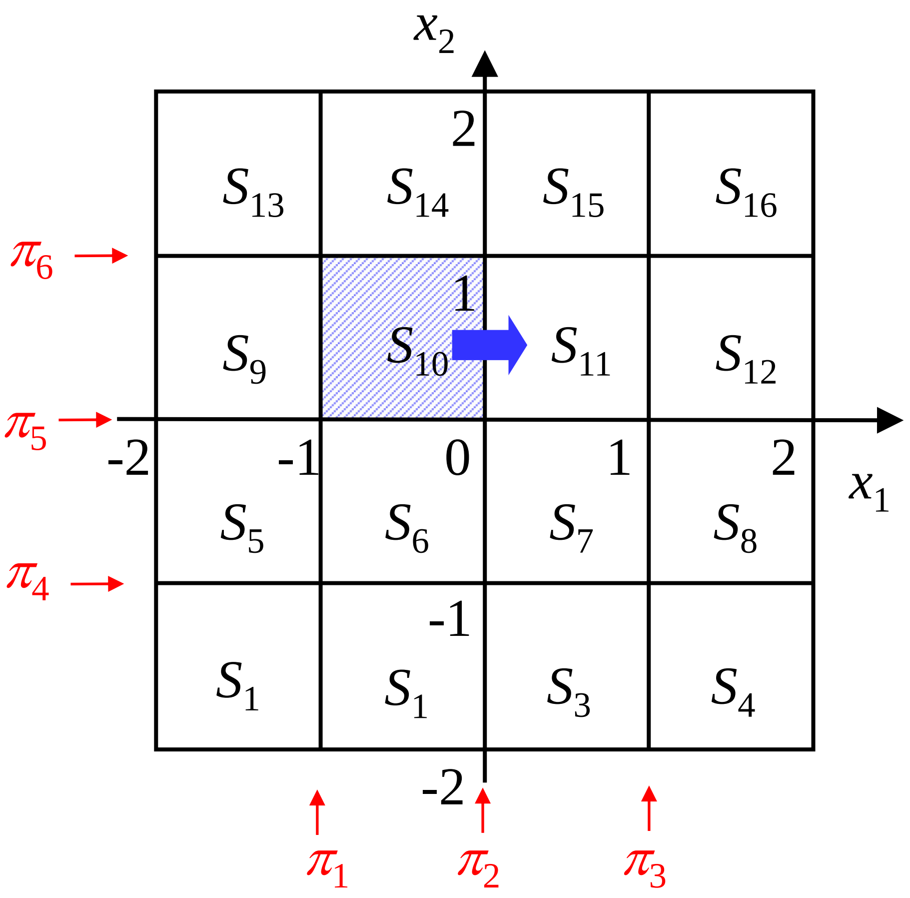
A PieceWise Linear System

Predicate abstraction: example (2)

Predicates $\Pi = \{ \pi_i \}$
$x_1 \le k\ (k = -1, 0, 1)$,
$x_2 \le k\ (k = -1, 0, 1)$.

Abstract states B
$S_1 = [1,1,1,1,1,1], \ldots,$
$S_{10} = [0,1,1,0,0,1], \ldots,$
$S_{16} = [0,0,0,0,0,0]$.

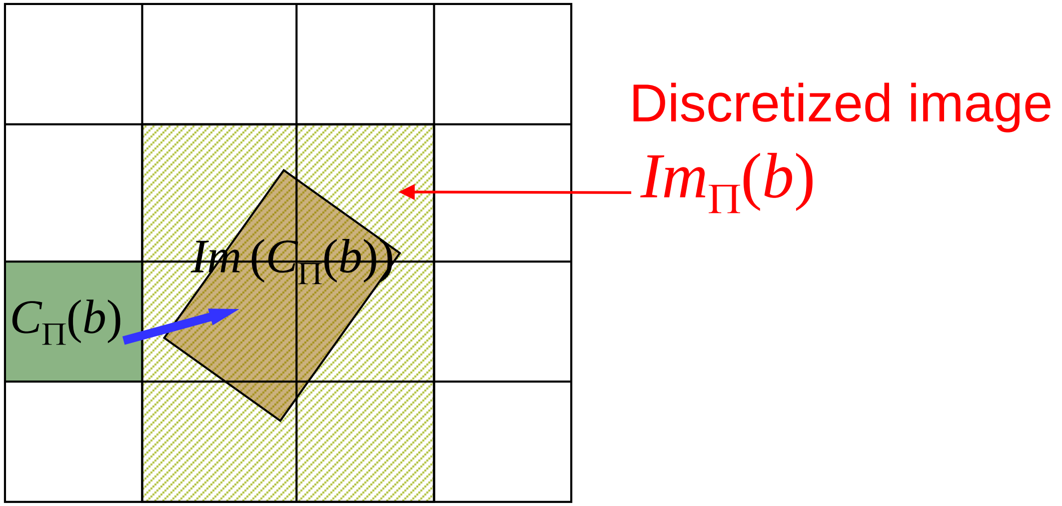
Transitions
$S_{10} \to S_{11} \equiv$
$\exists x \in S_{10}\ \exists x' \in S_{11}.\ x \to x'$
(over-approximation)



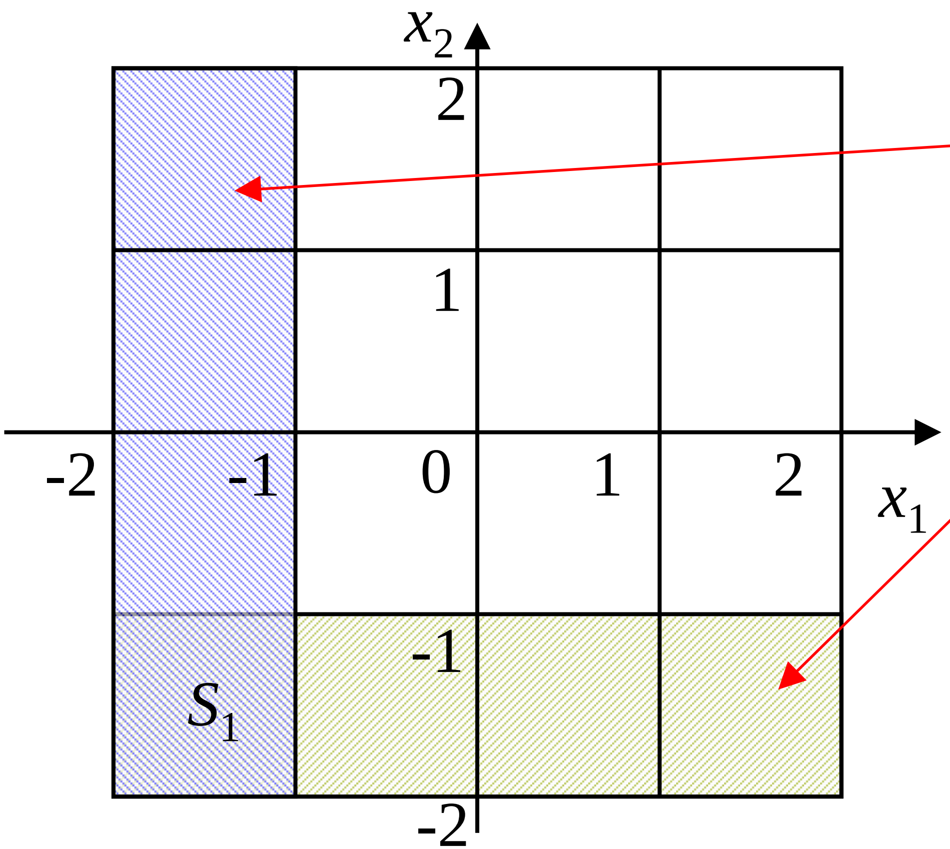
Exact computation

$b(v_1,\ldots,v_n)$: an abstract states with state variables $v_1,\ldots, v_n$
$C_\Pi(b) \subseteq \mathbf{R}^n$: Concretization of b
$\pi(R) \in \{0, 1\}^m$: Discretization of region R
$Im(R) \subseteq \mathbf{R}^n$: Image of region R
$Im_\Pi(b) := \pi(Im(C_\Pi(b)))$: Discretized Image of abstract state b

$$\Delta(v_1,\cdots,v_n,v_1',\cdots,v_n') := \bigvee_{b\in B} \left(b(v_1,\cdots,v_n) \wedge \mathrm{Im}_\Pi(b)(v_1',\cdots,v_n')\right)$$



Approximated computation (1)



Enlarged abstract states $\underline{B} = \{ \underline{b}_i \}$

$\underline{b}_1 = [1, 1, 1, dc, dc, dc]$

dc :don't care

$\underline{b}_2 = [dc, dc, dc, 1, 1, 1]$

$$\boxed{S_1 = \underline{b}_1 \wedge \underline{b}_2}$$

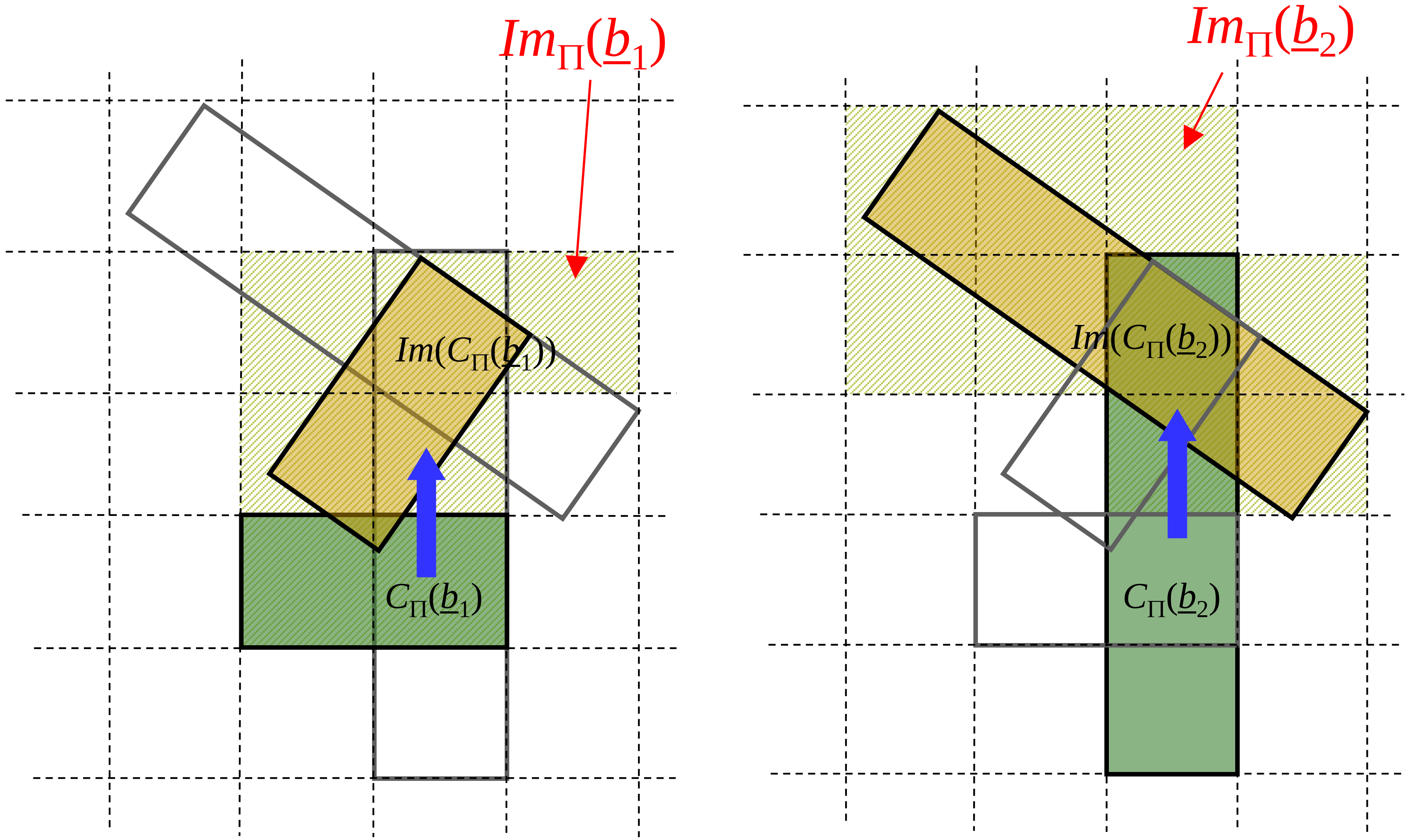
Conjunction complete:
Each abstract state is represented by conjunction of enlarged abstract states.

Approximated computation (2)

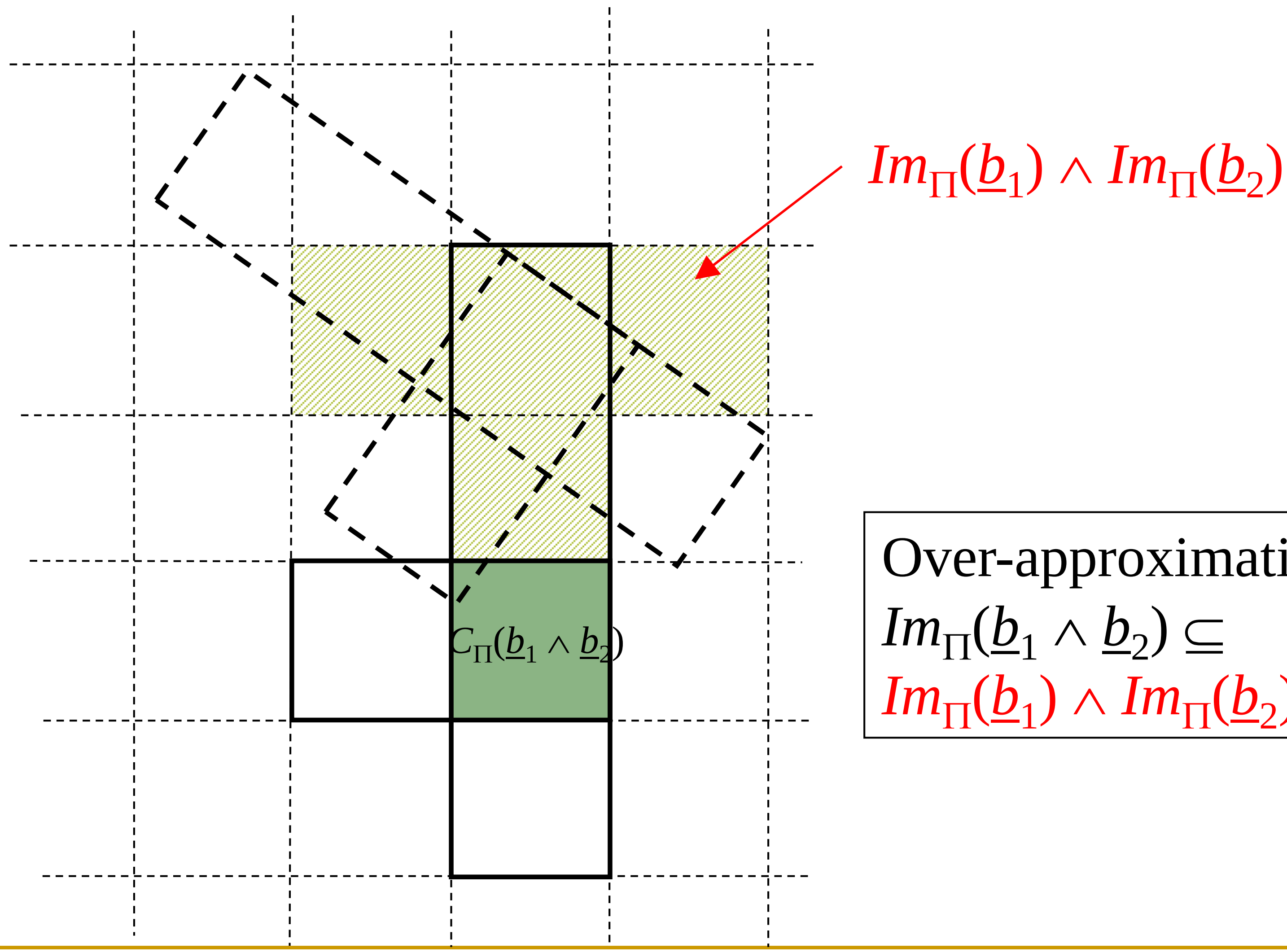
$\underline{b}(v_1,\ldots,v_n)$: enlarged abstract state with state variables $v_1,\ldots, v_n$
$\underline{B}$: the set of enlarged abstract states
$Im_\Pi(C_\Pi(\underline{b}))$: discretized image of $\underline{b}$

$$\Delta^{\sim}(v_1,\cdots,v_n,v_1',\cdots,v_n') := \bigwedge_{\underline{b}\in\underline{B}} \left(\underline{b}(v_1,\cdots,v_n) \to \mathrm{Im}_\Pi(\underline{b})(v_1',\cdots,v_n')\right)$$

Approximated computation (3)



Approximated computation (4)

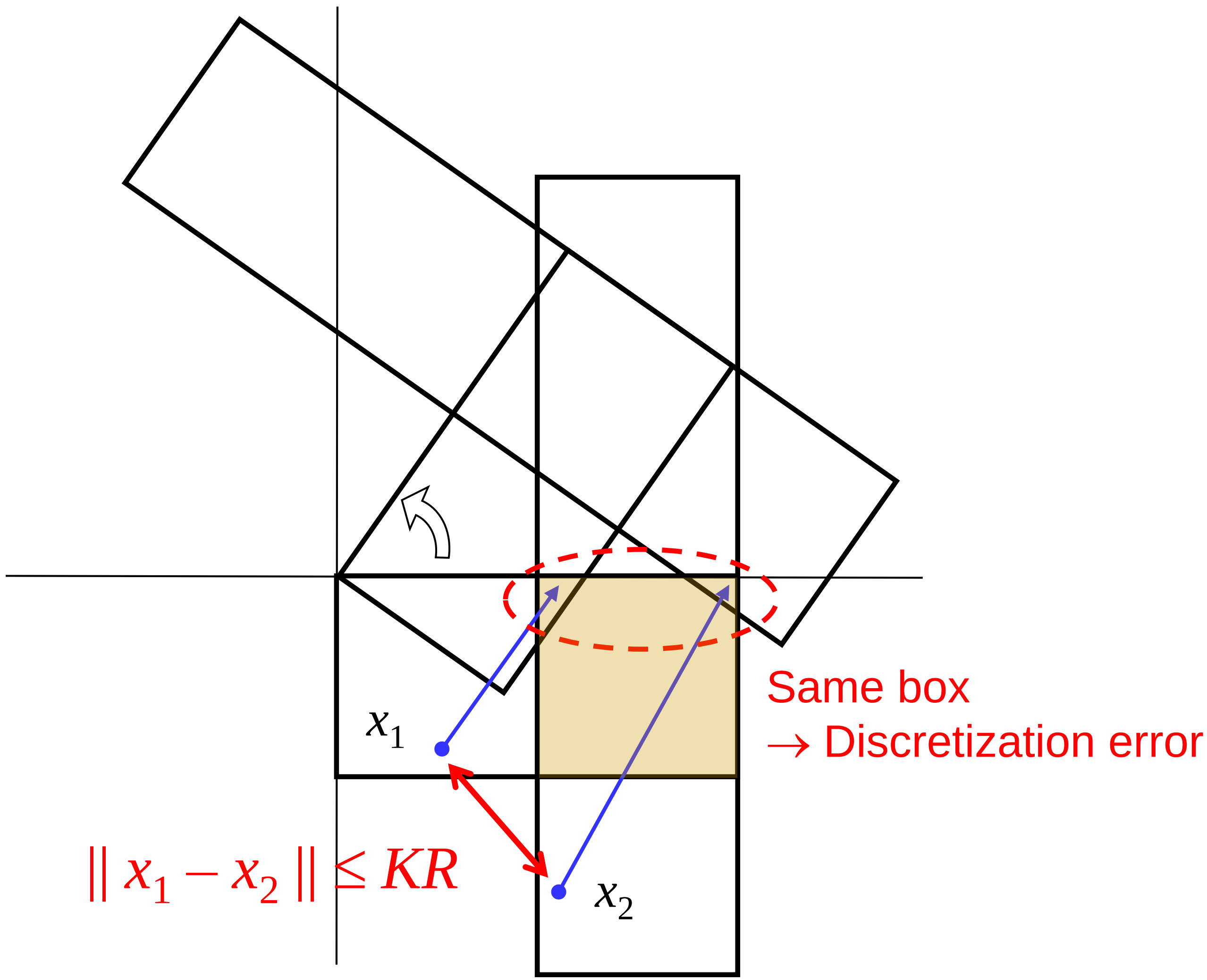


Over-approximation:
$Im_{\Pi}(\underline{b}_1 \wedge \underline{b}_2) \subseteq$
$Im_{\Pi}(\underline{b}_1) \wedge Im_{\Pi}(\underline{b}_2)$

Justification of the Idea (1)

- Discrete-time autonomous system: $x(t_{k+1}) = f(x(t_k))$.
- If f is injective (one-to-one), then $Im(Q_1 \cap Q_2) = Im(Q_1) \cap Im(Q_2)$. [This holds for discrete-time linear/affine systems.]
- Even if $Im(Q_1 \cap Q_2) = Im(Q_1) \cap Im(Q_2)$, $\pi(Im(Q_1 \cap Q_2) = \pi(Im_\Pi(Q_1)) \cap \pi(Im(Q_2))$ does not necessarily hold (discretization error).
- However, the error occurs *only around correct boxes*.
  - If $\| f^{-1}(x_1) - f^{-1}(x_2) \| / \| x_1 - x_2 \| \le K$ for any x_1, x_2 (Lipschitz continuity of f^{-1}), then $\| x_1 - x_2 \| \le K \| f(x_1) - f(x_2) \|$. [This holds for discrete-time linear/affine systems.]
  - Suppose that $x_1 \in Q_1 - Q_2$, $x_2 \in Q_2 - Q_1$, but $f(x_1)$ and $f(x_2)$ are in the same box.
  - Then, there exists a positive real R s.t. $\| f(x_1) - f(x_2) \| \le R$.
  - We have $\| x_1 - x_2 \| \le KR$.

Justification of the Idea (2)



Complexity

- h_i : the number of predicates in the *i*-th axis.
- The number of abstract states is

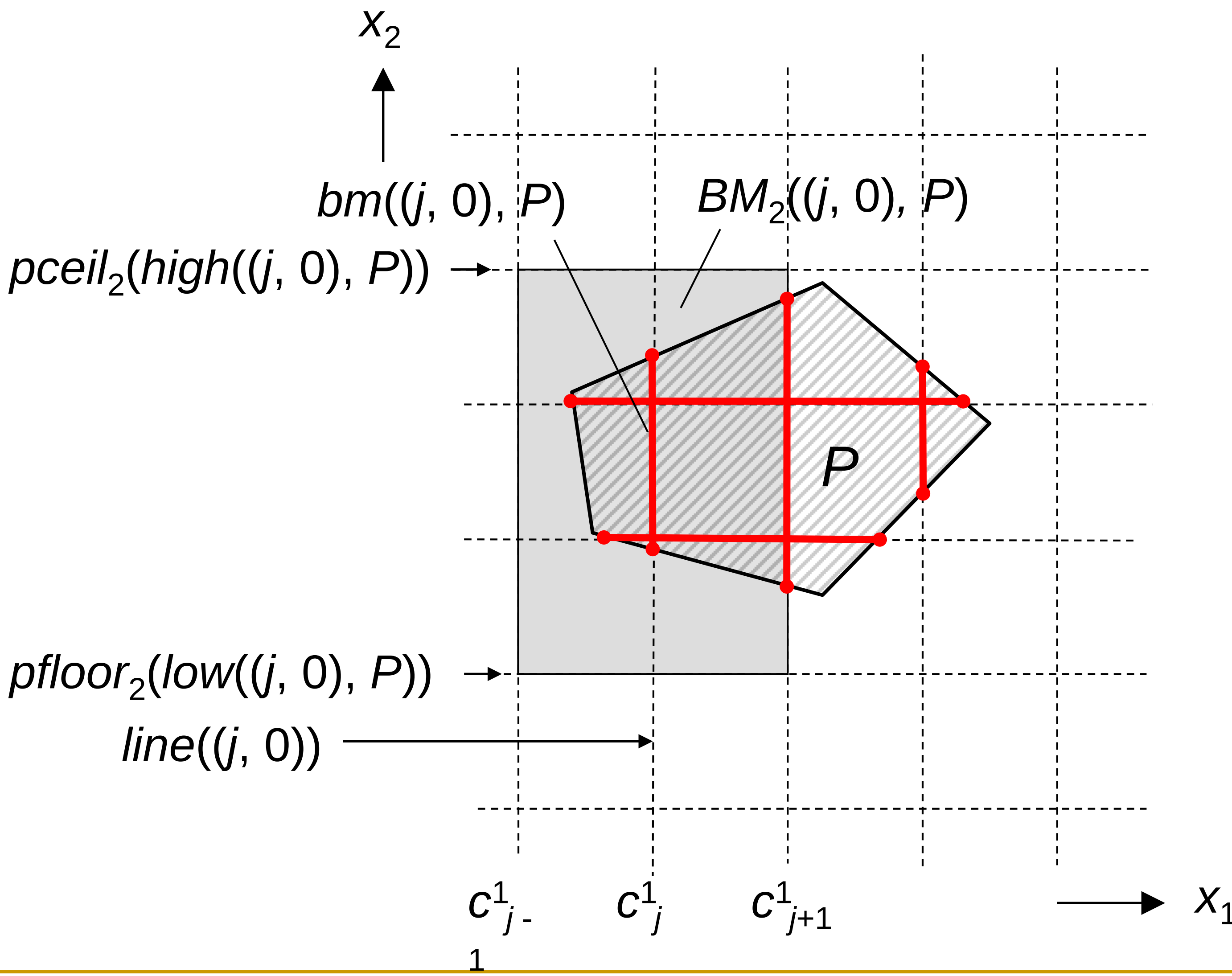
  $|B| = \Pi_{i = 1, n} (h_i + 1) = O((1 + m/n)^n)$.
- The number of enlarged abstract states is

  $|\underline{B}| = \Sigma_{i = 1, n} (h_i + 1) = O(m + n)$.

Discretization of Polyhedra: how to compute $Im_{\Pi}(\underline{b})$ from $Im(C_{\Pi}(\underline{b}))$

- Since $Im(C_{\Pi}(\underline{b}))$ is much larger than $Im(C_{\Pi}(b))$, the approximated computation requires more time at this step, provided that the computation time for the discretization *depends on the size of polyhedra*. As a result, the approximated computation is not very fast.
- We have develop an efficient algorithm, called *the beam method*, for this step. The algorithm uses convexity of regions.
- The beam method is compared with
  - Direct comparison : computing intersection between polyhedron *P* and each box in the axis-aligned bounding box of *P*,
  - Shannon expansion.

Beam Method for 2D Space



Systems with Inputs

$$x(t_{k+1}) = Ax(t_k) + Bu(t_k)$$

$$\Downarrow$$

$$\begin{pmatrix} x(t_{k+1}) \\ u \end{pmatrix} = \left(\begin{array}{c|c} A & B \\ \hline 0 & I \end{array}\right) \begin{pmatrix} x(t_k) \\ u \end{pmatrix}$$

We embed the input in the state space.
Then the matrix is nonsingular, provided that A is nonsingular.

Computation Results (1)

$$x(t_{k+1}) = Ax(t_k) + \begin{pmatrix} -0.1 \\ \vdots \\ -0.1 \end{pmatrix}$$

where A is an n-dimensional square matrix that represents the following rotations:
$n = 2$: $\pi/3$ around the origin.
$n = 3$: $\pi/3$ around the origin on x_{23}-plane, x_{13}-plane, and x_{12}-plane.
$n = 4$: $\pi/3$ around the origin on x_{12}-plane, x_{23}-plane, and x_{31}-plane.
$n = 5$: $\pi/3$ around the origin on x_{12}-plane, x_{23}-plane, x_{34}-plane, and x_{45}-plane.

Computation Results (2)

- τ_Π : the exact transitions.
- $\tau_\Pi{}^\sim$: the approximated transitions.
- Evaluation criteria:
  - Ratio $\gamma^\sim = |\ \tau_\Pi{}^\sim\ |\ /\ |\ \tau_\Pi\ |$.
  - The number of error transitions classified by hamming distance. Let $\#e_i$ be the number of transitions in $\tau_\Pi{}^\sim - H_i(\tau_\Pi)$, where $H_i(\tau_\Pi)$ is the set of all transitions whose *hamming distance* to at least one of transitions in τ_Π is no more than i.

Computation Results (3)

h	CPU time (sec.)			$\lvert\tau_{\Pi}\rvert$
	Direct	Shannon	Beam	
3	0.02	0.03	0.03	59
5	0.04	0.07	0.04	140
10	0.12	0.37	0.12	478
15	0.24	0.95	0.24	1,017
20	0.5	2.42	0.5	1,751
30	1.4	8.02	1.37	3,813
40	3.13	19.22	3.12	7,898
50	6.05	37.78	6.05	10,382
60	10.54	66.77	10.54	14,848
70	17.14	108.41	17.23	20,096
80	26.69	164.85	26.88	26,167
90	39.03	240.69	39.37	32,978
100	56.5	338.98	56.6	40,660

n = 2, Exact

h : the number of predicates in each axis

Computation Results (4)

h	CPU time (sec.)			$\lvert\tau_{\text{II}}^{\sim}\rvert$	$\gamma^{\sim}$	$\#e_1$
	Direct	Shannon	Beam			
3	0.02	0.02	0.01	67	1.14	0
5	0.04	0.06	0.02	162	1.16	0
10	0.14	0.56	0.07	555	1.16	0
15	0.29	0.23	0.11	1,190	1.17	0
20	0.72	1.4	0.17	2,060	1.18	0
30	2.26	4.57	0.38	4,488	1.18	0
40	5.42	10.88	0.73	7,898	1.18	0
50	10.53	21.34	1.29	12,252	1.18	0
60	18.67	37.57	2.13	17,530	1.18	0
70	30.4	60.76	3.16	23,718	1.18	0
80	46.02	91.79	4.7	30,913	1.18	0
90	67.13	133.34	6.45	38,943	1.18	0
100	95.2	188.06	9.01	48,037	1.18	0

n = 2, Approx.

Computation Results (5)

h	CPU time (sec.) Exact (Direct)	CPU time (sec.) Approx. (Beam)	$\lvert\tau_{\Pi}\rvert$	$\lvert\tau_{\Pi}^{\sim}\rvert$	$\gamma^{\sim}$	$\#e_1$	$\#e_2$
3	0.16	0.06	450	574	1.28	4	0
5	0.7	0.17	1,679	2,129	1.27	3	0
10	4.96	0.94	10,896	14,029	1.29	8	0
15	14.11	2.89	34,044	44,605	1.31	93	0
20	40.33	8.04	77,265	101,552	1.31	399	0
30	152.64	32.86	248,912	327,702	1.32	1,222	0

n = 3, Exact/Approx.

Computation Results (6)

h	CPU time (sec.)		$\lvert\tau_\Pi\rvert$	$\lvert\tau_\Pi^\sim\rvert$	$\gamma^\sim$	$\#e_1$	$\#e_2$	$\#e_3$
	Exact (Direct)	Approx. (Beam)						
3	3.67	0.56	3,927	6,095	1.55	71	0	0
5	15.08	2.45	22,322	36,673	1.64	468	0	0
7	43.91	7.25	72,957	123,526	1.69	1,738	0	0
10	196.55	28.32	265,111	459,607	1.73	7,050	1	0

n = 4, Exact/Approx.

Computation Results (7)

h	CPU time (sec.)		$\lvert\tau_\Pi\rvert$	$\lvert\tau_{\tilde{\Pi}}\rvert$	$\gamma^\sim$	$\#e_1$	$\#e_2$	$\#e_3$
	Exact (Direct)	Approx. (Beam)						
2	10.98	1.65	6,233	10,997	1.76	412	0	0
3	44.04	5.19	32,338	61,282	1.90	2,294	8	0
4	172.24	14.08	110,703	220,908	2.00	9,697	62	0
5	443.49	32.02	284,518	578,721	2.03	26,034	150	0
7	-	128.88	-	-	-	-	-	-
10	-	692.89	-	-	-	-	-	-

n = 5, Exact/Approx.

Future Work

- Application to parameter design of hybrid dynamical systems.
- Development of method for general predicates.