

Hands on PESSOA

`http://www.cyphylab.ee.ucla.edu`

`http://code.google.com/p/pessoa`

Supported Specifications

1. $\Box C$
2. $\Diamond T$
3. $\Box C \wedge \Diamond T$
4. $\Box C \wedge \Diamond \Box T$

Three step procedure

1. Compute abstraction

>>**pss_abstract**

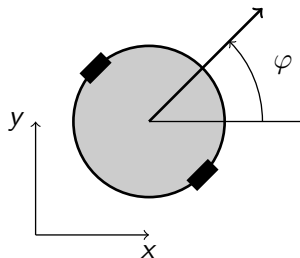
2. Compute predicates C , T

>>**pss_build_set**

3. Solve synthesis problem

>>**pss_design**

Example



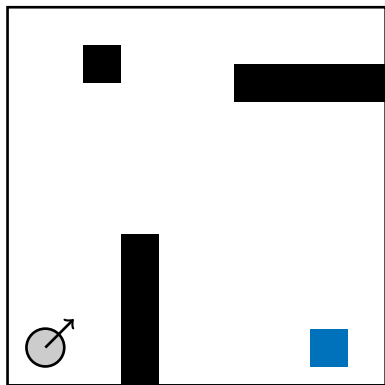
■ dynamics

$$\dot{x}(t) = v \cos \varphi(t)$$

$$\dot{y}(t) = v \sin \varphi(t)$$

$$\dot{\varphi}(t) = u$$

Example



- dynamics

$$\dot{x}(t) = v \cos \varphi(t)$$

$$\dot{y}(t) = v \sin \varphi(t)$$

$$\dot{\varphi}(t) = u$$

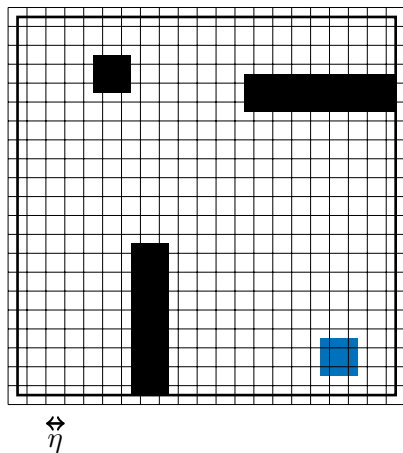
- state space

$$X = [0, 4] \times [0, 4] \times [-\pi, \pi]$$

- input space

$$U = [0, 1] \times [-1, 1]$$

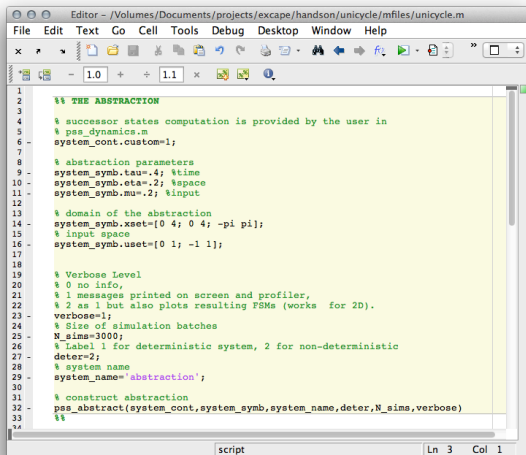
1. Compute abstraction



Parameters

- sampling time $\tau = 0.4$
- state space discretization $\eta = 0.2$
- input space discretization $\mu = 0.2$

1. Compute abstraction



```
1 %% THE ABSTRACTION
2
3
4 % successor states computation is provided by the user in
5 % pss_dynamics.m
6 system_cont.custom=1;
7
8 % abstraction parameters
9 system_symb.tau=.4; %time
10 system_symb.eta=.2; %space
11 system_symb.mu=.2; %input
12
13 % domain of the abstraction
14 system_symb.xset=[0 4; 0 4; -pi pi];
15 % input space
16 system_symb.uset=[0 1; -1 1];
17
18
19 % Verbose Level
20 % 0 no info,
21 % 1 messages printed on screen and profiler,
22 % 2 as 1 but also plots resulting FSMs (works for 2D).
23 verbose=1;
24 % Size of simulation batches
25 N_sims=3000;
26 % Label 1 for deterministic system, 2 for non-deterministic
27 deter=2;
28 % system name
29 system_name='abstraction';
30
31 % construct abstraction
32 pss_abstract(system_cont,system_symb,system_name,deter,N_sims,verbose)
33 %%
34
```

pss_abstract

- `system_cont.custom=1`
Successor states are computed in `pss_dynamics.m`
- sampling time $\tau = 0.4$
- state space discretization $\eta = 0.2$
- input space discretization $\mu = 0.2$
- state space $X = [0, 4] \times [0, 4] \times [-\pi, \pi]$
- input space $U = [0, 1] \times [-1, 1]$

output

abstraction.bdd

abstraction_symb.mat

2. Compute predicates

```
Editor - /Volumes/Documents/projects/escape/handson/unicycle/mfiles/unicycle.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
x
1.0 1.1
36  %% PREDICATES
37
38
39  %% THE TARGET SET
40
41  % compute BDD of the target set
42  % Construct_type:
43  % 0 - using hyperboxes,
44  % 1 - using pss_target_set.m
45  construct_type=0;
46
47  %target_set = [2 4; 0 1; -system1_symb.eta system1_symb.eta];
48  target_set = [1.5 2.5; 0 .5; -0.25 0.25];
49
50  % set filename
51  Tset='target';
52
53  %compute predicate
54  pss_build_set(system_name,target_set,Tset,N_sims,construct_type,verbose)
55
56
57  %% THE CONSTRAINT SET
58
59  % compute BDD of the constraint set
60  % Construct_type:
61  % 0 - using hyperboxes,
62  % 1 - using pss_target_set.m
63  construct_type=1;
64  % set filename
65  Cset='constraints';
66
67  pss_build_set(system_name,system_symb.xset,Cset,N_sims,construct_type,verbose)
68  %%
69
70
script Ln 37 Col 1
```

pss_build_set

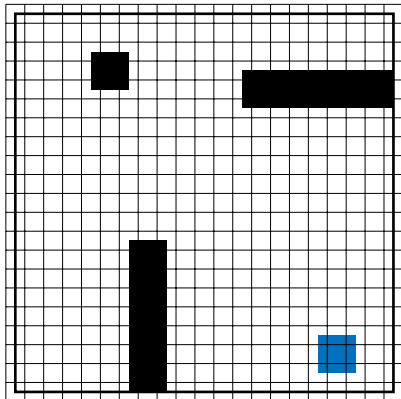
- type=0
define hyper-cube
- type=1
characteristic function of safe
set defined in
[pss_target_set.m](#)

output

constraints.bdd
constraints_symb.mat

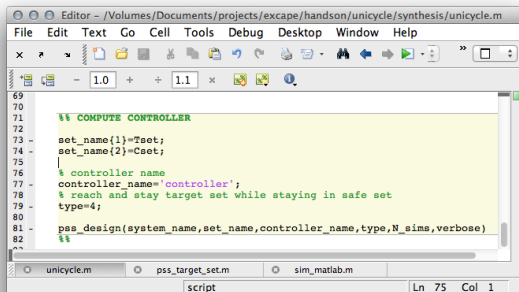
2. Compute predicates

pss_target_set.m



```
Editor - /Volumes/Documents/projects/escape/handson/unicycle/synthesis/pss_target_set.m
File Edit Text Go Cell Tools Debug Desktop Window Help
x [Navigation icons] [Zoom: 1.0] [Zoom: 1.1] [Icons]
1 function add_state = pss_target_set(x)
2
3     add_state = pss_target_set(x)
4
5     % This function allows the user to define any characteristic function
6     % in any fashion to define the target set, as long as
7     % the state x (input) can be applied. Return {0,1} based on whether
8     % the state belongs to the target set or not.
9
10    %     INPUTS: x - state to be considered [x1;x2;...]
11    %
12    %     OUTPUT: add_sate - Return {0,1} depending on whether the state
13    %             passed belongs to characteristic function
14    %
15    %     Anna Davitian <davitian@ee.ucla.edu>,  CyPhyLab-UCLA 2009
16
17    % The user is free to define any characteristic function
18
19    % default: whole state space is safe
20    add_state=1;
21
22    if x(1)>=1.2 && x(1)<=1.6 && x(2)<=1.6
23        add_state=0;
24    end
25
26    if x(1)>=2.4 && x(2)>=3 && x(2)<=3.4
27        add_state=0;
28    end
29
30    if x(1)>=.8 && x(1)<=1.2 && x(2)>=3.2 && x(2)<=3.6
31        add_state=0;
32    end
33
```

3. Solve synthesis problem



```
69
70
71 %% COMPUTE CONTROLLER
72
73 set_name(1)=Tset;
74 set_name(2)=Cset;
75 |
76 % controller name
77 controller_name='controller';
78 % reach and stay target set while staying in safe set
79 type=4;
80
81 pss_design(system_name,set_name,controller_name,type,N_sims,verbose)
82 %%
```

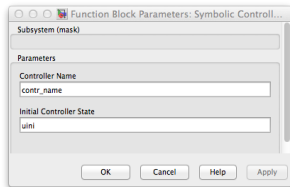
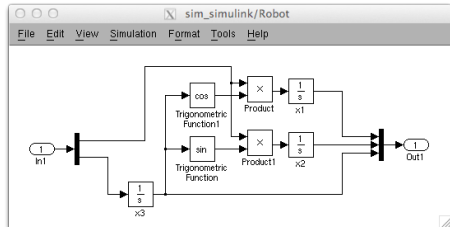
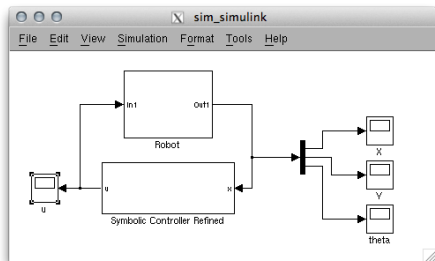
pss_design

- type=1
 $\square C$
- type=2
 $\diamond T$
- type=3
 $\square C \wedge \diamond T$
- type=4
 $\square C \wedge \diamond \square T$

output

controller.bdd
controller_dom.bdd
controller_symb.mat

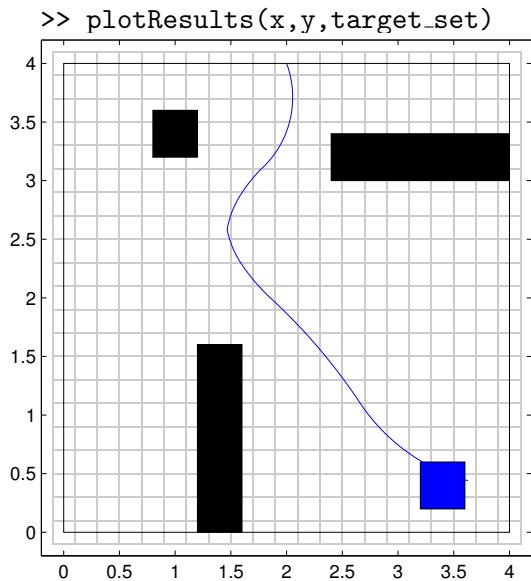
Simulation in Simulink



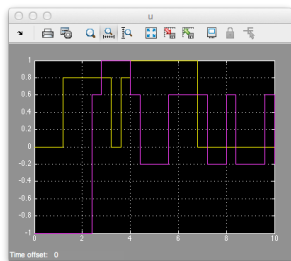
```
>> sim_simulink
>> contr_name='controller'
>> xini=[0;0.5;0]
>> uini=[0;0]
```

Simulink saves x,y,phi and u to the MATLAB workspace

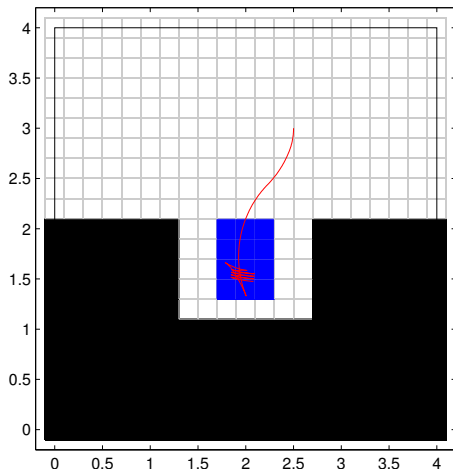
Visualization



Simulink scope: v, u



Parallel parking



Constraints

Turning $u \neq 0$ is allowed only if $v \neq 0$!

Pre-computed abstraction (files):
abstraction_parallel.bdd

Update:

- `system_name='abstraction_parallel'`
- `target_set`
- `pss_target_set.m`

Appendix

1. Compute abstraction

`pss_dynamics(params_symb, x, u)`

