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**

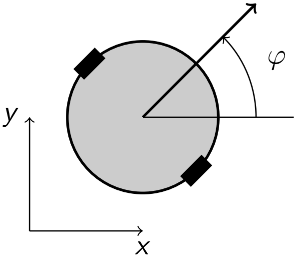
2. Compute predicates C, T

   >>**pss_build_set**

3. Solve synthesis problem

   >>**pss_design**

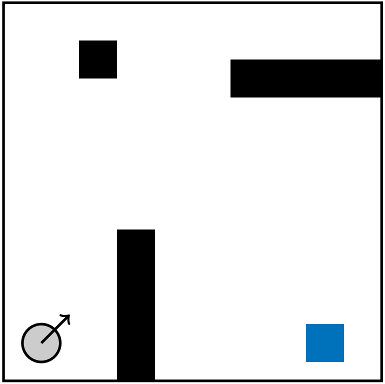
Example



- dynamics

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

Example

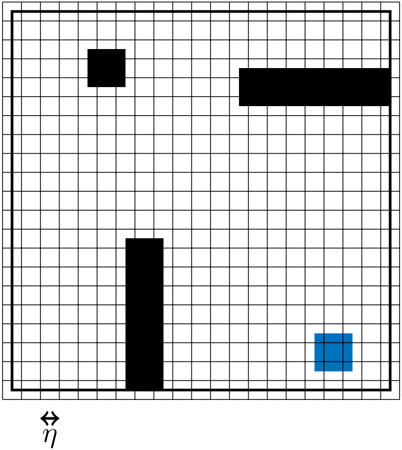


- dynamics

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

- state space
  $X = [0, 4] \times [0, 4] \times [-\pi, \pi]$
- input space
  $U = [0, 1] \times [-1, 1]$

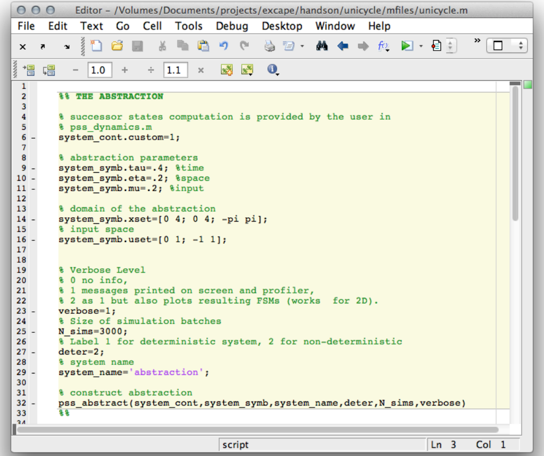
1. Compute abstraction



Parameters

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

1. Compute abstraction



pss_abstract

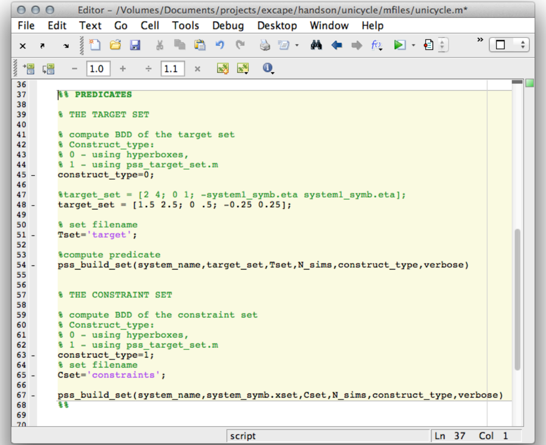
- `system_cont.custom=1` Successor states are computed in `pss_dynamics.m`
- sampling time $\tau = 0.4$
- state space discretization $\eta = 0.2$
- input space discretization $\mu = 0.2$
- state space $X = [0,4] \times [0,4] \times [-\pi, \pi]$
- input space $U = [0,1] \times [-1,1]$

output

`abstraction.bdd`

`abstraction_symb.mat`

2. Compute predicates



pss_build_set

- type=0
  define hyper-cube
- type=1
  characteristic function of safe set defined in
  `pss_target_set.m`

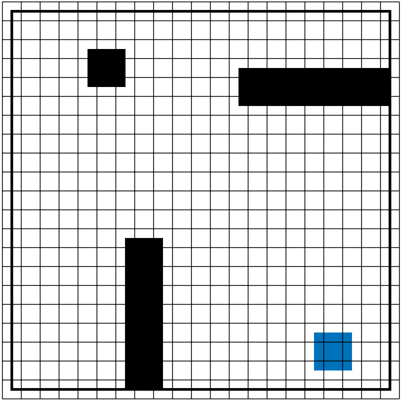
output

`constraints.bdd`

`constraints_symb.mat`

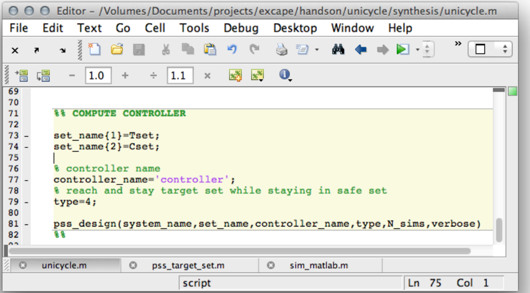
2. Compute predicates

`pss_target_set.m`



```
function add_state = pss_target_set(x)
%
%   add_state = pss_target_set(x)
%
%   This function allows the user to define any characteristic function
%   in any fashion to define the target set, as long as
%   the state x (input) can be applied. Return {0,1} based on whether
%   the state belongs to the target set or not.
%
%       INPUTS: x - state to be considered [x1;x2;...]
%
%       OUTPUT: add_sate - Return {0,1} depending on whether the state
%               passed belongs to characteristic function
%
%   Anna Davitian <davitian@ee.ucla.edu>,   CyPhyLab-UCLA 2009

% The user is free to define any characteristic function

% defaul: whole state space is safe
add_state=1;

if x(1)>=1.2 && x(1)<=1.6 && x(2)<=1.6
    add_state=0;
end

if x(1)>=2.4 && x(2)>=3 && x(2)<=3.4
    add_state=0;
end

if x(1)>=.8 && x(1)<=1.2 && x(2)>=3.2 && x(2)<=3.6
    add_state=0;
end
```

3. Solve synthesis problem



pss_design

- type=1
  $\Box C$
- type=2
  $\Diamond T$
- type=3
  $\Box C \wedge \Diamond T$
- type=4
  $\Box C \wedge \Diamond \Box T$

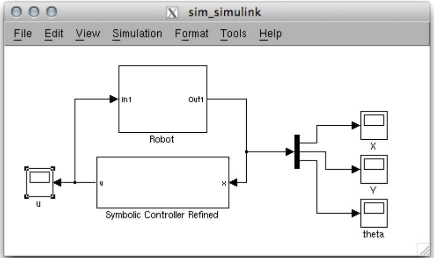
output

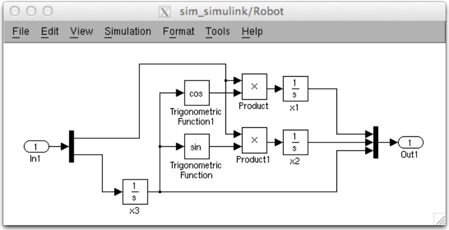
controller.bdd

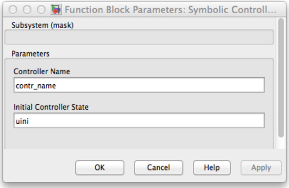
controller_dom.bdd

controller_symb.mat

Simulation in Simulink





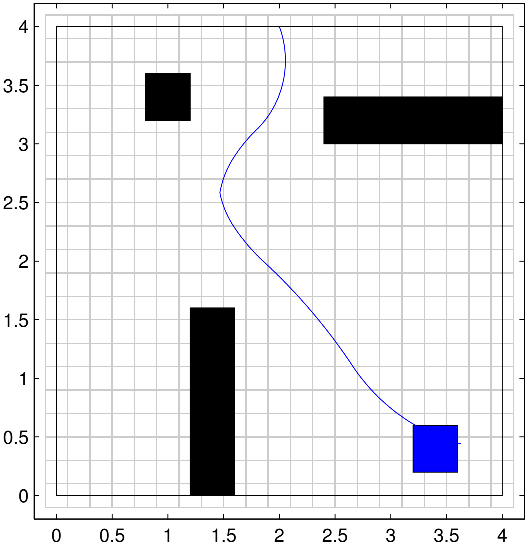



```
>> sim_simulink
>> contr_name='controller'
>> xini=[0;0.5;0]
>> uini=[0;0]
```

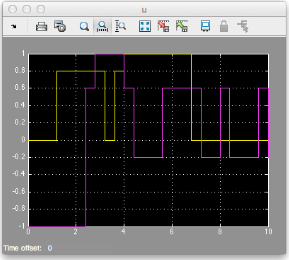
Simulink saves `x,y,phi` **and** `u` **to the MATLAB workspace**

Visualization

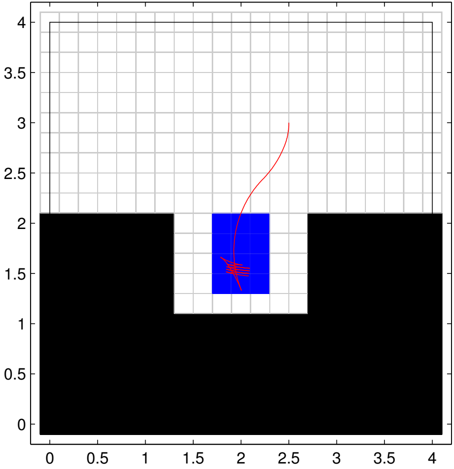
```
>> plotResults(x,y,target_set)
```



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)
```