Parametric Statistical Model Checking of UAV Flight Plan

Ran BAO[1], Christian Attiogbé[2], Paulin Fournier[2] and Didier Lime[3]

[1] PIXIEL Group / LS2N UMR CNRS 6004, Nantes, France

[2] Université de Nantes / LS2N UMR CNRS 6004, Nantes, France

[3] Central Nantes / LS2N UMR CNRS 6004, Nantes, France

MSR 2019

Summary

PIXIEL LS2N UNIVERSITÉ DE NANTES

Outline

PIXIEL LS2N UNIVERSITÉ DE NANTES

Motivation

UAVs flying above a crowd (Entertainment)

⇒ **How to ensure that the flight is secure ?**

PIXIEL LS2N UNIVERSITÉ DE NANTES

Contributions

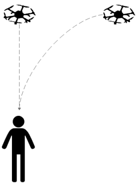
We propose a model of the UAV system

- In the context of a flight plan
- Parametric : takes into account
  - Sensor/Filter precision and failure
  - Wind force
- Allows to predict the trajectory

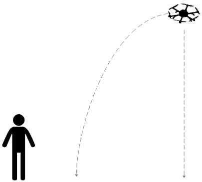
We propose and use parametric statistical model checking techniques

- Computes an approximation of the probability of satisfying a property
  - as a parametric function
  - polynomial
  - with parametric confidence intervals
- Experimentations with industrial case study

Position of drone security concerns



Bad position



Good position

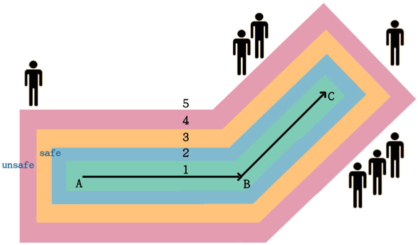
calculate probability of being in the good/bad position

$\Rightarrow$ **What is the good position ?**

PIXIEL LS2N UNIVERSITÉ DE NANTES

Safety zone

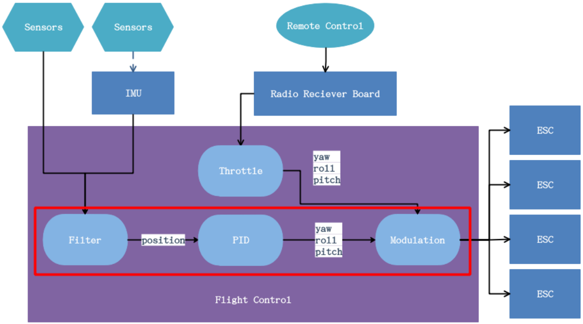
5 zones inline with avionic certification(DO-178C)



- Take account flight plan and components
- Fixed size : depended application
- critical zones : 4 , 5

Main components of an UAV

Physical components + Software

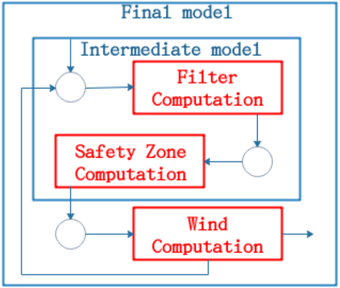


- Position estimation = Sensors + filter
- Stabilize computation = PID
- Parameter = Precision of (sensors + filter)

PIXIEL LS2N UNIVERSITÉ DE NANTES

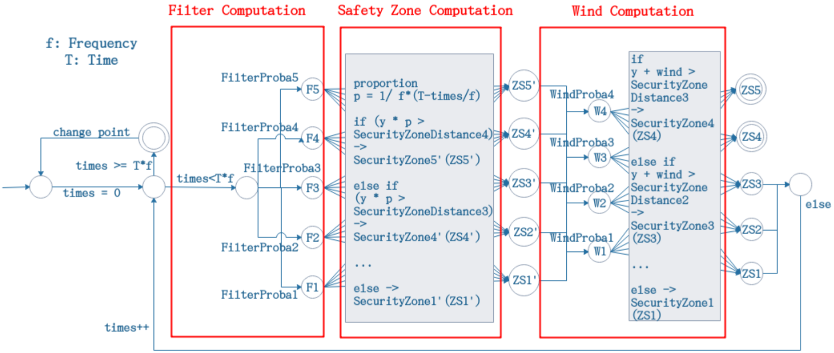
The proposed approach

A method to build and verify UAV model

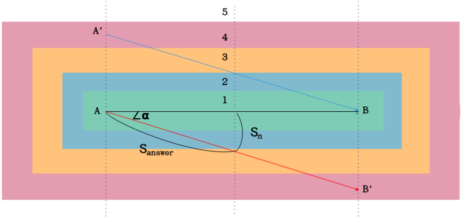


- Filter capacity in parameters
- Computation filter effect
- Add rotate effect (angles in the trajectory)
- Add wind effect (additional parameter of the model)

Resulting model



Importance of time on the deviation



The estimated position (A') impacts on the target.
The time impact (T_{answer})

$$Sn = \sin\alpha * S_{answer} \quad (1)$$

$$\sin\alpha = \frac{AA'}{A'B} = \frac{AA'}{\sqrt{AA'^2 + AB^2}} \quad (2)$$

$$S_{answer} = V * T_{answer} \quad (3)$$

(V : UAV's velocity)

Outline

Building the model : Markov Chains (MC)

A Markov Chain is a purely probabilistic model $\mathcal{M} = (S, s_0, P)$, where
S is a set of states,
$s_0 \in S$ is the initial state,and
$P : S \times S \mapsto [0, 1]$ is a probabilistic transition function that,
given a pair of states (s_1, s_2), yields the probability of moving from s_1 to s_2.

Definitions :

- Finite run : $\rho = s_0 s_1 \ldots s_n$ s.t. $P(s_i, s_{i+1}) > 0$
- $\Gamma(l)$: set of all runs of length l in $\mathcal{M}$
- Probability of finite run : $\rho = s_0 s_1 \ldots s_n$, $\mathbb{P}_{\mathcal{M}}(\rho) = \prod_{i=1}^{n} P(s_{i-1}, s_i)$

PIXIEL LS2N UNIVERSITÉ DE NANTES

Building the model : Parametric Markov Chain (pMC)

A pMC is a tuple $\mathcal{M} = (\mathcal{S}, s_0, P, \mathbb{X})$ such that
$\mathcal{S}$ is a finite set of states,
$s_0 \in \mathcal{S}$ is the initial state,
$\mathbb{X}$ is a finite set of parameters, and
$P : \mathcal{S} \times \mathcal{S} \mapsto Poly(\mathbb{X})$ is a parametric transition probability function, expressed as a polynomial on $\mathbb{X}$.

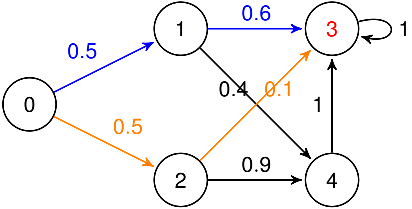
If $v \in \mathbb{R}^{\mathbb{X}}$ is a valuation of the parameters,

- P_v : transition probabilities under v : $P_v(s, s') = P(s, s')(v)$
- v is valid if $(\mathcal{S}, s_0, P_v)$ is a MC
- $\mathcal{M}^v = (\mathcal{S}, s_0, P_v)$
- Runs and probabilities are similar to MC, but parametric

Our formal model of the UAV is built using a parametric Markov chain.
Now, we need to check our model.

PIXIEL LS2N UNIVERSITÉ DE NANTES

Basis for statistic model checking : Monte Carlo for MCs



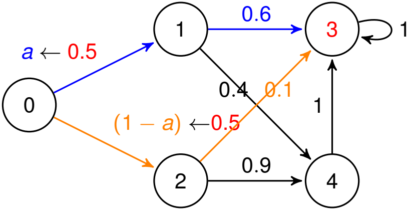
1 $\rho_1 = 0 \to 2 \to 4 \quad R(\rho_1) = 0$
2 $\rho_2 = 0 \to 1 \to 3 \quad R(\rho_2) = 1$
3 $\rho_3 = 0 \to 2 \to 4 \quad R(\rho_3) = 0$
4 $\rho_4 = 0 \to 1 \to 4 \quad R(\rho_4) = 0$
5 $\rho_5 = 0 \to 1 \to 4 \quad R(\rho_5) = 0$
6 $\rho_6 = 0 \to 1 \to 3 \quad R(\rho_6) = 1$
7 $\rho_7 = 0 \to 2 \to 3 \quad R(\rho_7) = 1$
8 $\rho_8 = 0 \to 2 \to 4 \quad R(\rho_8) = 0$

- Run n simulations ρ_i of length l. (here $n = 8$ and $l = 2$)
- $r(\rho_i) = 1$ if ρ_i reaches 3 in two steps
- $\mathbb{E}^l_{\mathcal{M}}(r) \sim \frac{\sum r(\rho_i)}{n} \Rightarrow$ Here, $\mathbb{E}^2_{\mathcal{M}}(r) \sim \frac{3}{8} = 0.375$ $(exact : 0.35)$

Expected reward $\mathbb{E}^l_{\mathcal{M}}(r)$ is the expected value of r on the runs of length l.

UNIVERSITÉ DE NANTES

Basis for statistic model checking : Monte Carlo for pMCs

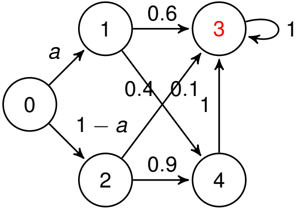


1 $\rho_1 = 0 \rightarrow 2 \rightarrow 4 \quad R(\rho_1) = 0$
2 $\rho_2 = 0 \rightarrow 1 \rightarrow 3 \quad R(\rho_2) = 0.6a$
3 $\rho_3 = 0 \rightarrow 2 \rightarrow 4 \quad R(\rho_3) = 0$
4 $\rho_4 = 0 \rightarrow 1 \rightarrow 4 \quad R(\rho_4) = 0$
5 $\rho_5 = 0 \rightarrow 1 \rightarrow 4 \quad R(\rho_5) = 0$
6 $\rho_6 = 0 \rightarrow 1 \rightarrow 3 \quad R(\rho_6) = 0.6a$
7 $\rho_7 = 0 \rightarrow 2 \rightarrow 3 \quad R(\rho_7) = 0.1(1-a)$
8 $\rho_8 = 0 \rightarrow 2 \rightarrow 4 \quad R(\rho_8) = 0$

- Use a normalization function $f \rightarrow \mathcal{M}^f$
- $R(\rho_i) = \mathbb{P}_{\mathcal{M}}(\rho)$ if ρ_i reaches 3 in two steps, 0 otherwise
- $\mathbb{E}^I_{\mathcal{M}}(r) = \mathbb{E}(\sum_{i=1}^{n}(\frac{R(\rho_i)}{\mathbb{P}_{\mathcal{M}^f}(\rho_i)})/n)(v)$
- Here, $\mathbb{E}^2_{\mathcal{M}}(r') \sim 0.275a + 0.25$ (exact : 0.5a+0.1)
- For v(a)=0.6 : $\mathbb{E}^2_{\mathcal{M}}(r') \sim 0.415$ (exact : 0.4)

UNIVERSITÉ DE NANTES

Parametric Statistical Model Checking (IMCpMC) [1]



For 500 runs, we get :
$\mathbb{E}^3_{\mathcal{M}}(r') \sim 0.592 * a + 0.092 \sim 0.4552$

For 5000 runs, we get :
$\mathbb{E}^3_{\mathcal{M}}(r') \sim 0.516 * a + 0.092 \sim 0.4016$

(exact : 0.5a+0.1 $\sim$ 0.4)
(ps : $v(a) = 0.6$)

```python
import model_old
from sympy import symbols

a = symbols('a')
Length=2

class Example(model_old.AbstractPMC):
    def getLength(self):
        return Length

    def initial(self):
        return [0,0]#[state, times]

    def next(self, a_state):
        s,t=a_state
        l1 =[0]
        l2=[[0,0]]

        if s==0:
            l1 = [a,1-a]
            l2 = [[1,t+1],[2,t+1]]
        elif s==1:
            l1 = [0.6,0.4]
            l2 = [[3,t+1],[4,t+1]]
        elif s==2:
            l1 = [0.1,0.9]
            l2 = [[3,t+1],[4,t+1]]
        elif s==3:
            l1 = [1]
            l2 = [[3,t+1]]
        elif s==4:
            l1 = [1]
            l2 = [[3,t+1]]

        return l1,l2

    def end(self, a_state):
        print(a_state)
        s,t = a_state
        return t==Length, s==3
```

- PRISM : with filter
- PARAM : with filter on parameter
- Parametric Statistical Model Checking (Python) : IMCpMC

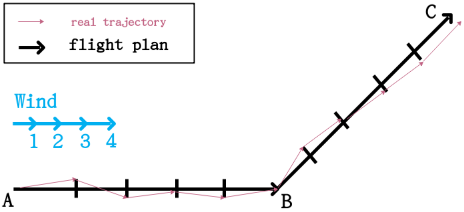
PIXIEL · LS2N · UNIVERSITÉ DE NANTES

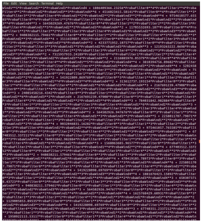
1. Available at https ://github.com/Astlo/IMCpMC

Outline

PIXIEL LS2N UNIVERSITÉ DE NANTES

Advantage of polynomial





```
senario:  1 Time= 1 , f= 1 , NbSim= 100000 , > 80 , for(0, 1 )
Computation duration (s):  115.70776295661926
0.04766666666666667*ProbaFilter3*ProbaWind1 + 0.12566666666666667*ProbaFilter3*Prob
aWind2 + 0.1984*ProbaFilter3*ProbaWind3 + 0.277*ProbaFilter3*ProbaWind4 + 0.9444
99999999999*ProbaFilter4*ProbaWind1 + 0.8676666666666666*ProbaFilter4*ProbaWind2
+ 0.812*ProbaFilter4*ProbaWind3 + 0.7320000000000001*ProbaFilter4*ProbaWind4
```

- Which probability is more present ?
- Wind effect

Experimentation

Results interpretation

	Model	10k		20k		50k	
		V1	**V2**	**V1**	**V2**	**V1**	**V2**
Running time	A	28s		51-54s		142-143s	
Scenario 1	A	4.99%	5.09%	4.74%	5.10%	4.91%	4.98%
Conf. interv.	A	±0.85%	±0.82%	±0.55%	±0.56%	±0.36%	±0.37%
Running time	B	28s		53-54s		149-155s	
Scenario 1	B	5.44%	5.31%	5.61%	5.21%	5.59%	5.47%
Conf. interv.	B	±0.98%	±0.86%	±0.69%	±0.64%	±0.42%	±0.43%
Running time	C	185-190s		311-314s		612-621s	
Scenario 1	C	4.95%	5.97%	5.28%	6.62%	4.16%	5.61%
Conf. interv.	C	±5.22%	±5.71%	±4.71%	±6.25%	±1.86%	±4.38%

model A : Filters as parameters
model B : Wind as parameter (number) but not present in polynomial
model C : Parameter filter and wind all in polynomial

Summary and future work

Summary :

- Formal model of UAV flight plan
- Parametric safety analysis

- Parametric Monte Carlo procedure for pMC
- Polynomial parametric confidence interval
- Prototype implementation

Future work :

- Experimentation and implementation improvements
- Change the direction of the wind

PIXIEL LS2N UNIVERSITÉ DE NANTES

Thanks

Thank you for your attention

Any questions ?