INTERNATIONAL CAE CONFERENCE

17th - 18th OCTOBER 2016

PARMA
PAGANINI CONGRESSI
ITALY

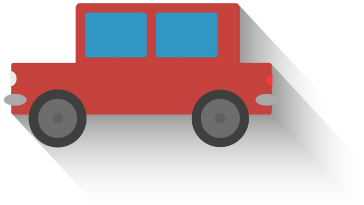
Validation of Automotive Control Applications using Formal Methods and metamodeling techniques

- Simone Silvetti, Esteco Spa & University Udine
- Mariapia Marchi, Esteco Spa

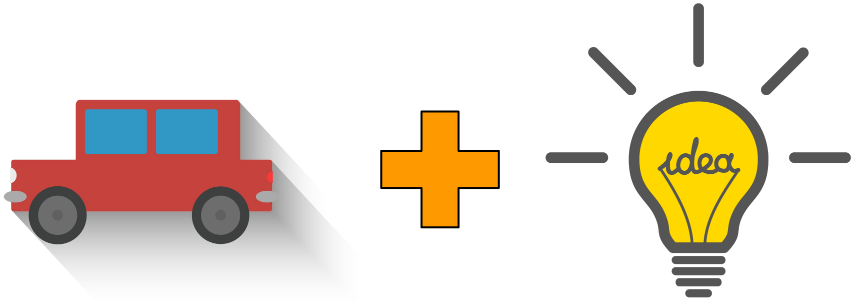
MDB (Model Based Development)

- process aimed at designing complex systems
- cost reduction
- reduce development time

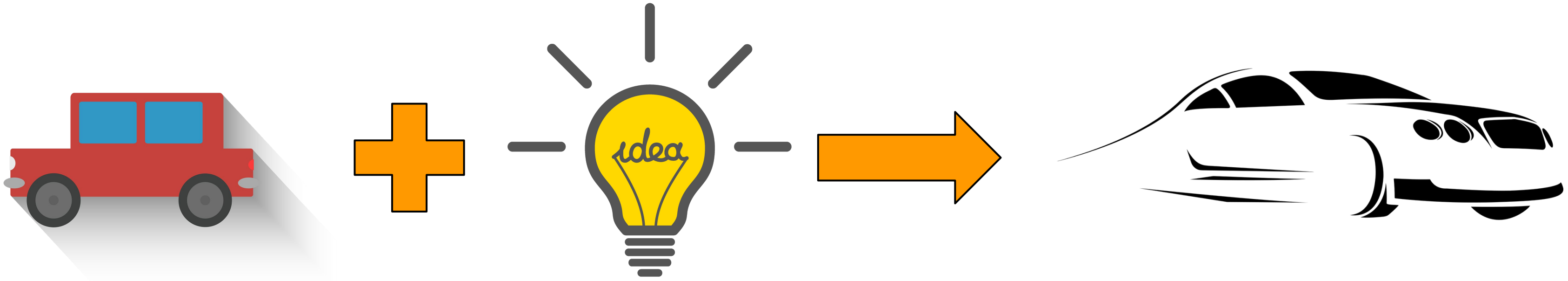
MDB (Model Based Development)



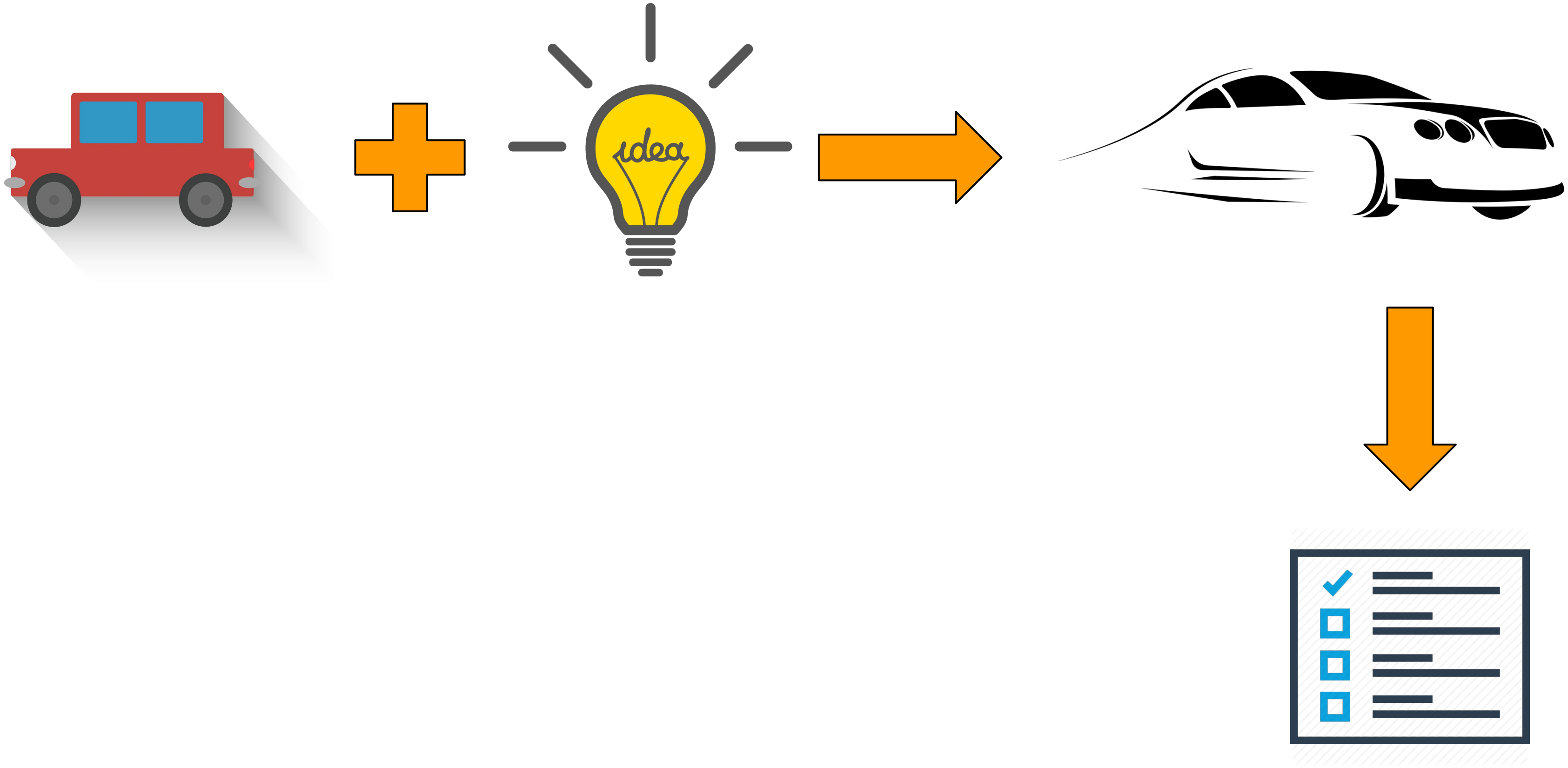
MDB (Model Based Development)



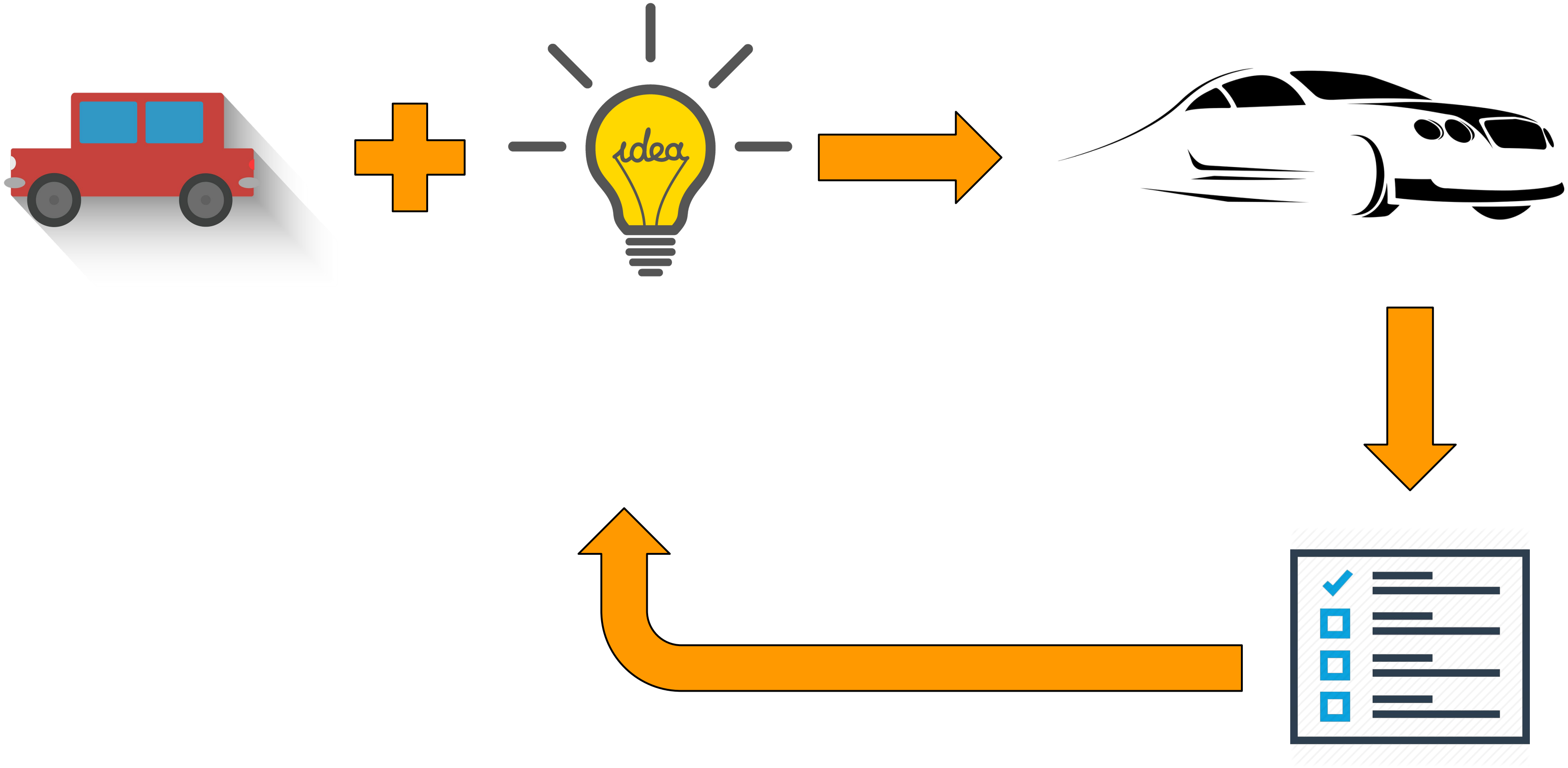
MDB (Model Based Development)



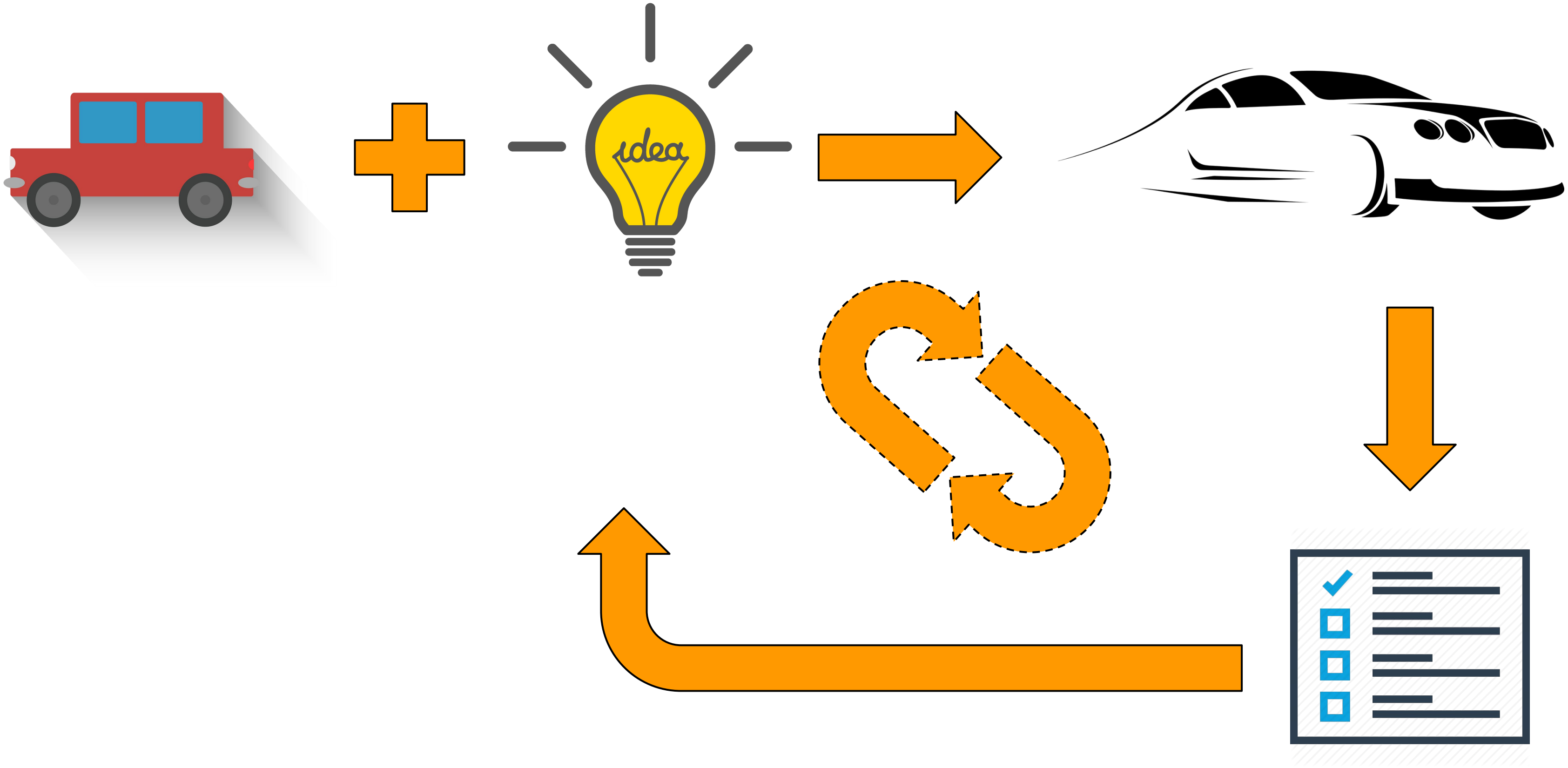
MDB (Model Based Development)



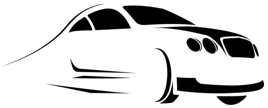
MDB (Model Based Development)



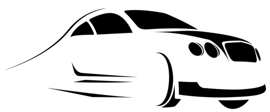
MDB (Model Based Development)



Validation Process



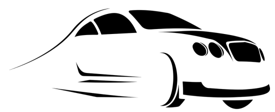
Validation Process



❖ Use of block diagram tools (Simulink, Gt suite)

❖ Powerful Tools but complex

Validation Process



- ❖ Use of block diagram tools (Simulink, Gt suite)
- ❖ Powerful Tools but complex

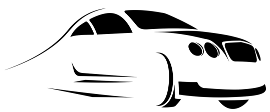
- ❖ Use of natural languages
- ❖ Involves time events...
- ❖ Not rigorous
- ❖ Not Machine interpretable

Validation Process

- ❖ Use of block diagram tools (Simulink, Gt suite)
- ❖ Powerful Tools but complex

- ❖ Use of natural languages
- ❖ Involves time events...
- ❖ Not rigurous
- ❖ Not Machine interpretable

Validation Process

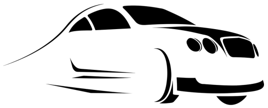


- ❖ Use of block diagram tools (Simulink, Gt suite)
- ❖ Powerful Tools but complex

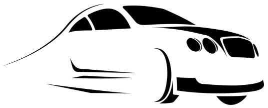
- ❖ Use of natural languages
- ❖ Involves time events…
- ❖ Not rigurous
- ❖ Not Machine interpretable

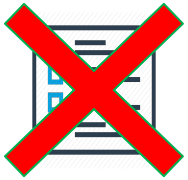
FORMAL METHODS !

Validation Process



Validation Process

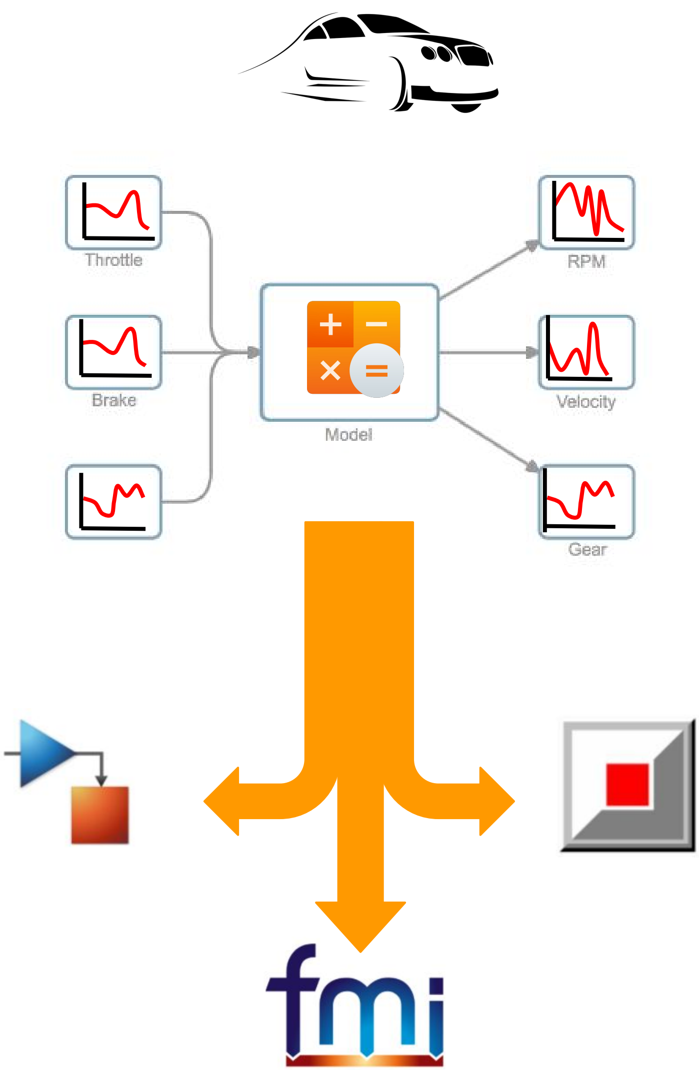




Validation Process

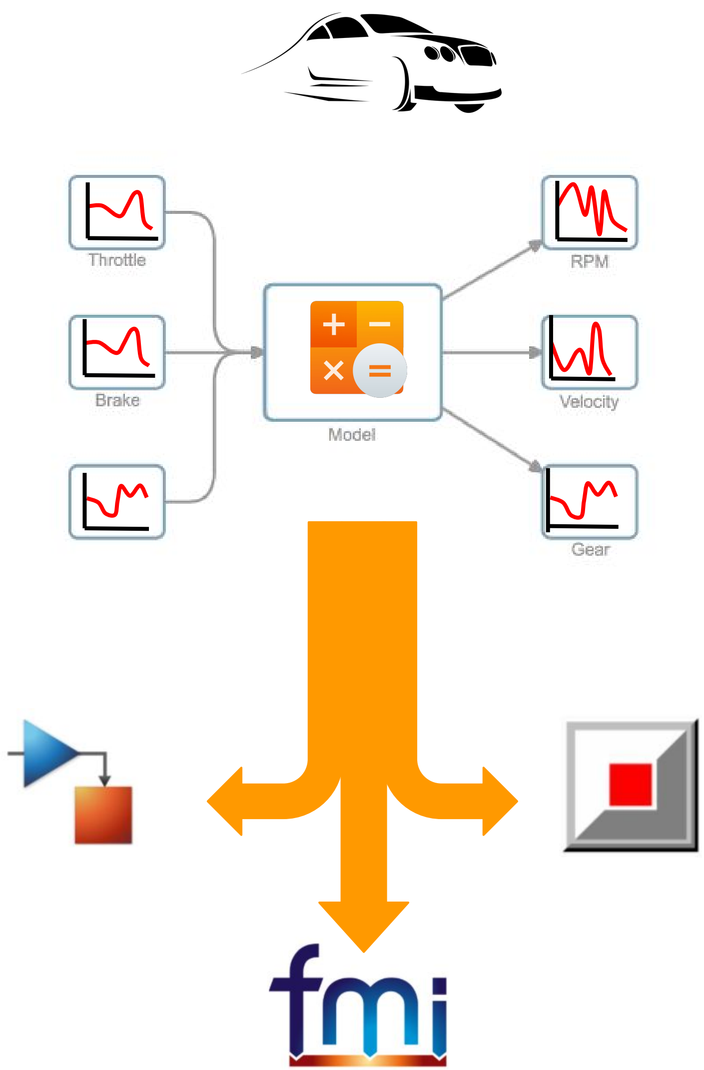
φ

Validation Process



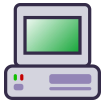
φ

Validation Process



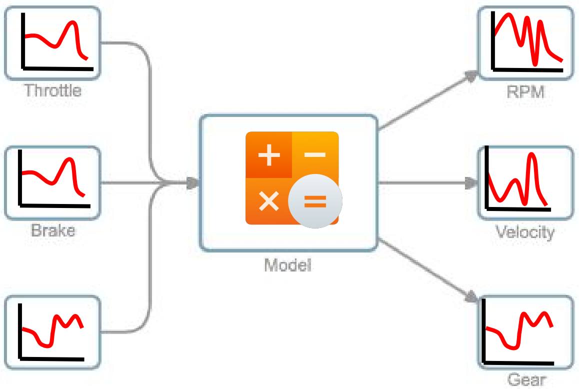
φ

"If the engine speed (w) is always less than k_1 then vehicle speed (v) can not exceed k_2 in less than T sec"



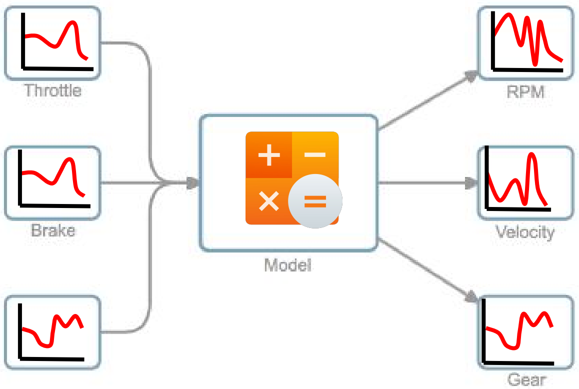
$$\neg(\mathbf{F}_{[0,T]}(v \geq k_2) \wedge \mathbf{G}(w \leq k_1))$$

Robustness Semantics



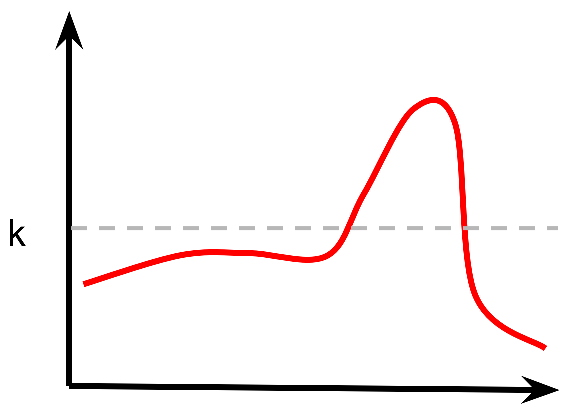
$\models \varphi$?

Robustness Semantics



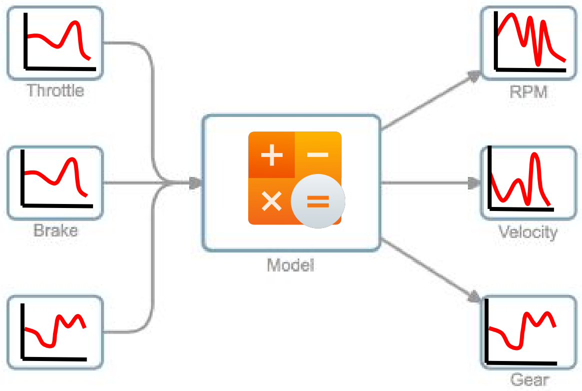
⊨ φ ?

F(f>k)

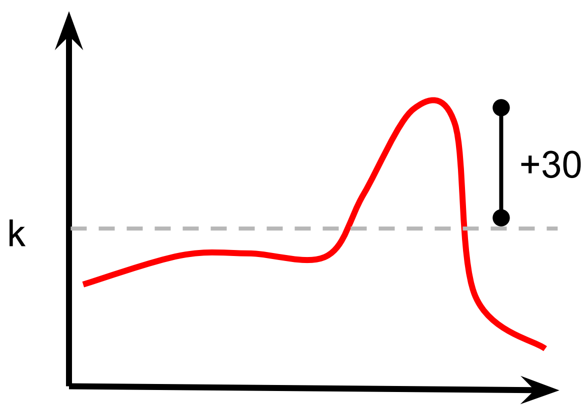


Boolean	
yes/no	

Robustness Semantics



$\models \varphi$?

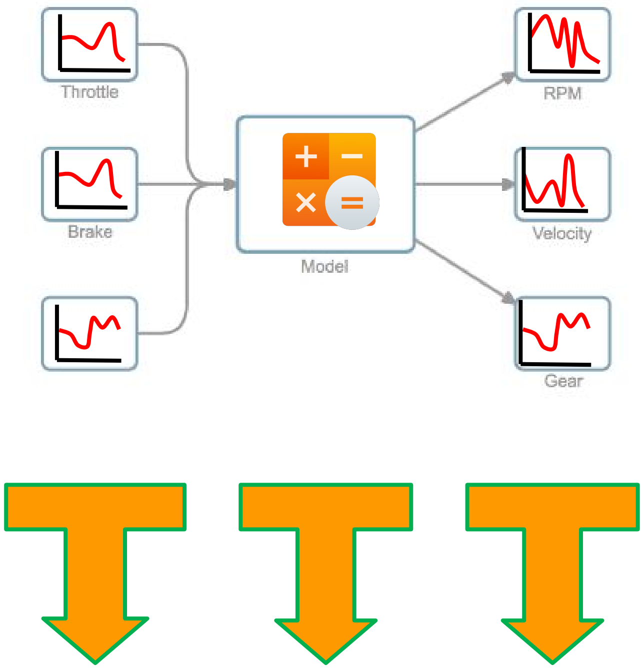


F(f>k)

Boolean	**Robustness**
yes/no	+30 / -30

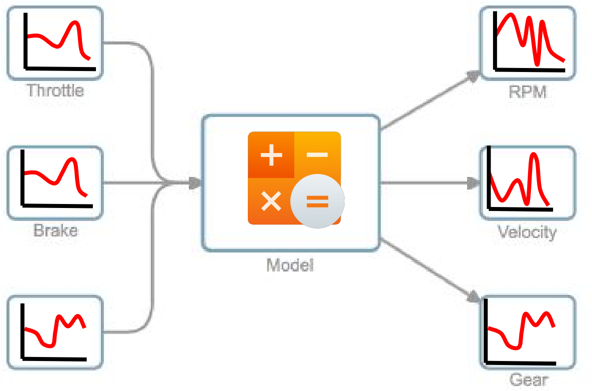
More Information!

The goal



$$f \longrightarrow M \longrightarrow M(f)$$

The goal

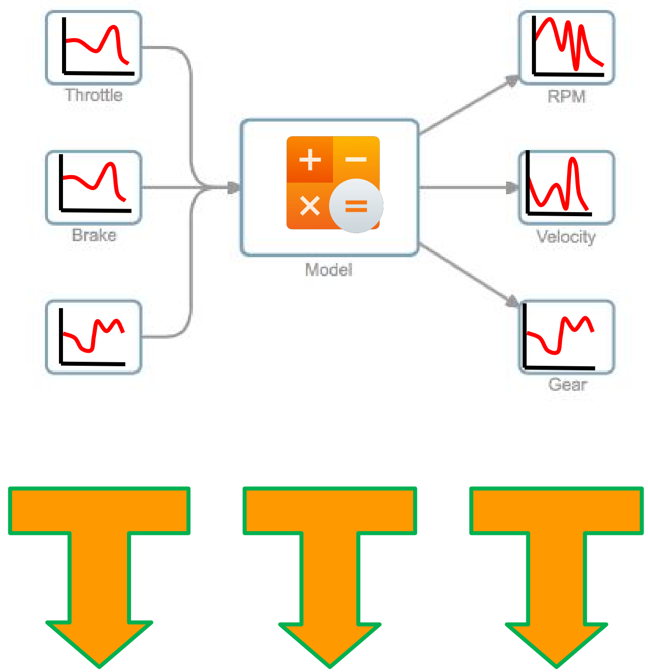


$$f \longrightarrow M \longrightarrow M(f)$$

The optimization Problem

$$R = \min_{f \in F} \; [M(f), \varphi]$$

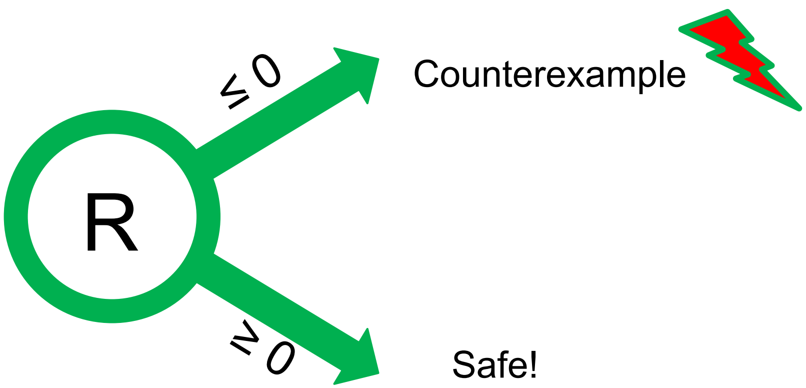
The goal



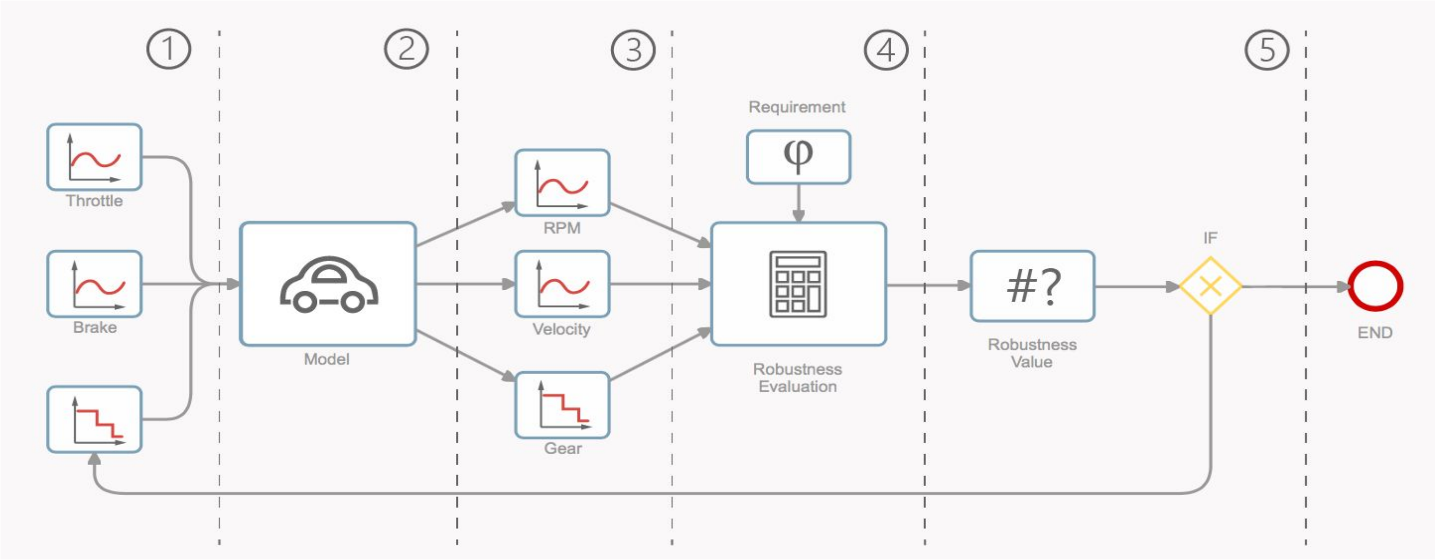
$f \longrightarrow M \longrightarrow M(f)$

The optimization Problem

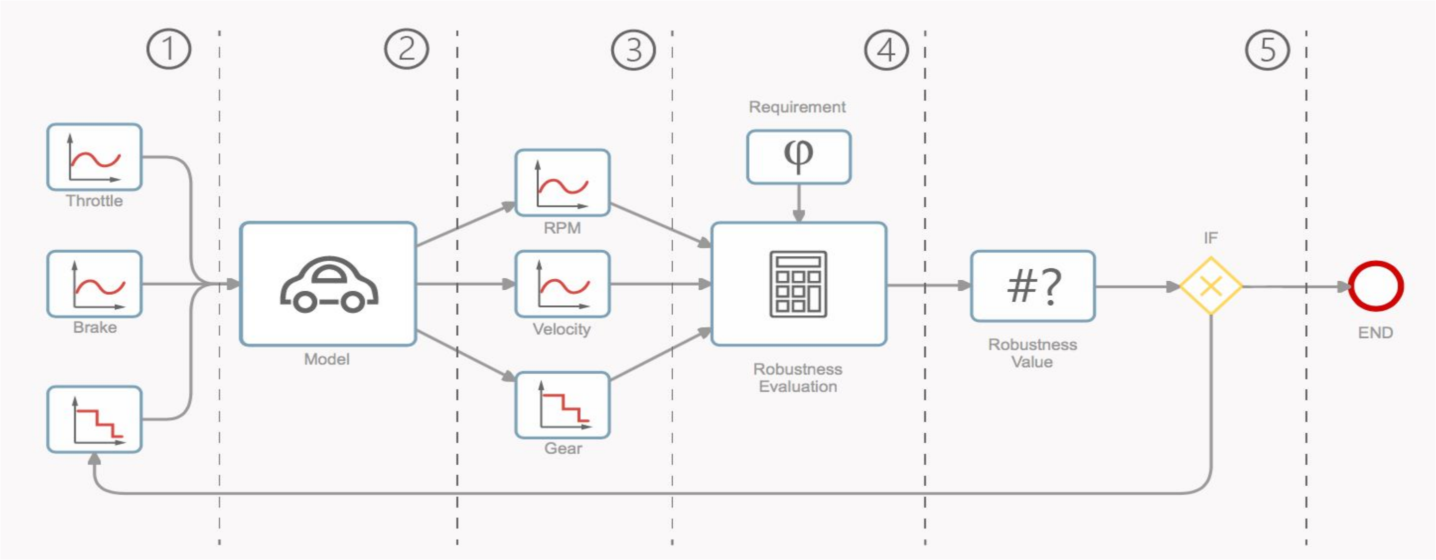
$$R = \min_{f \in F} \; [M(f), \varphi]$$



The optimization process



The optimization process

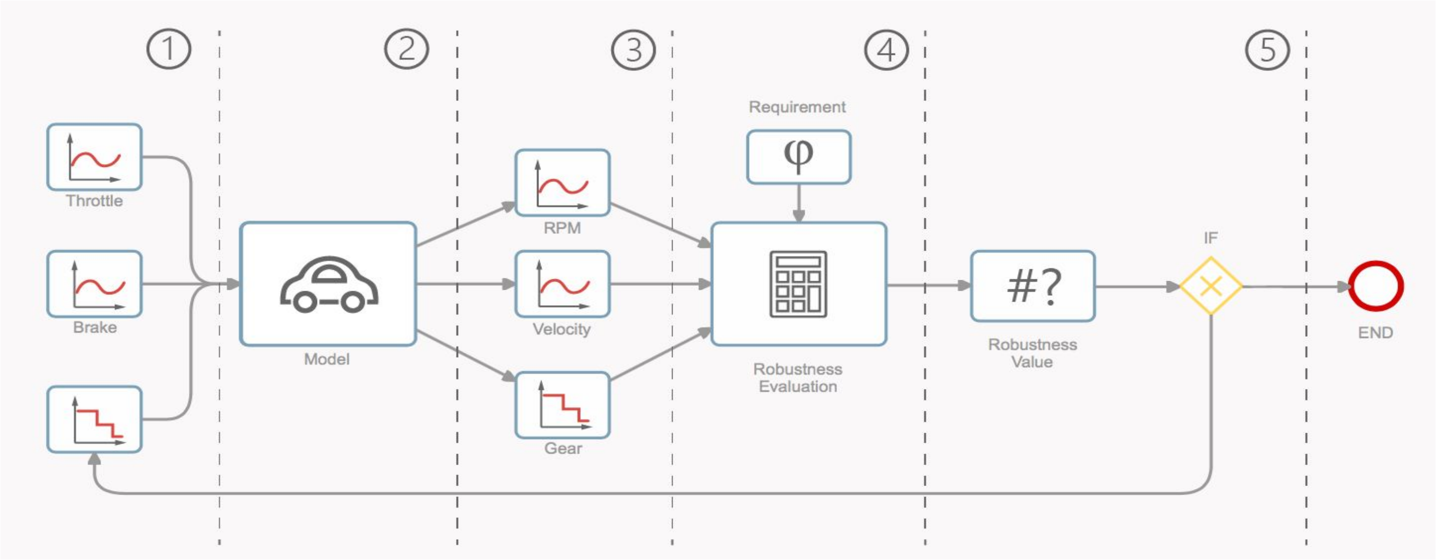


Challenges

- Low number of model execution

- Inputs are functions (temporal series)!!

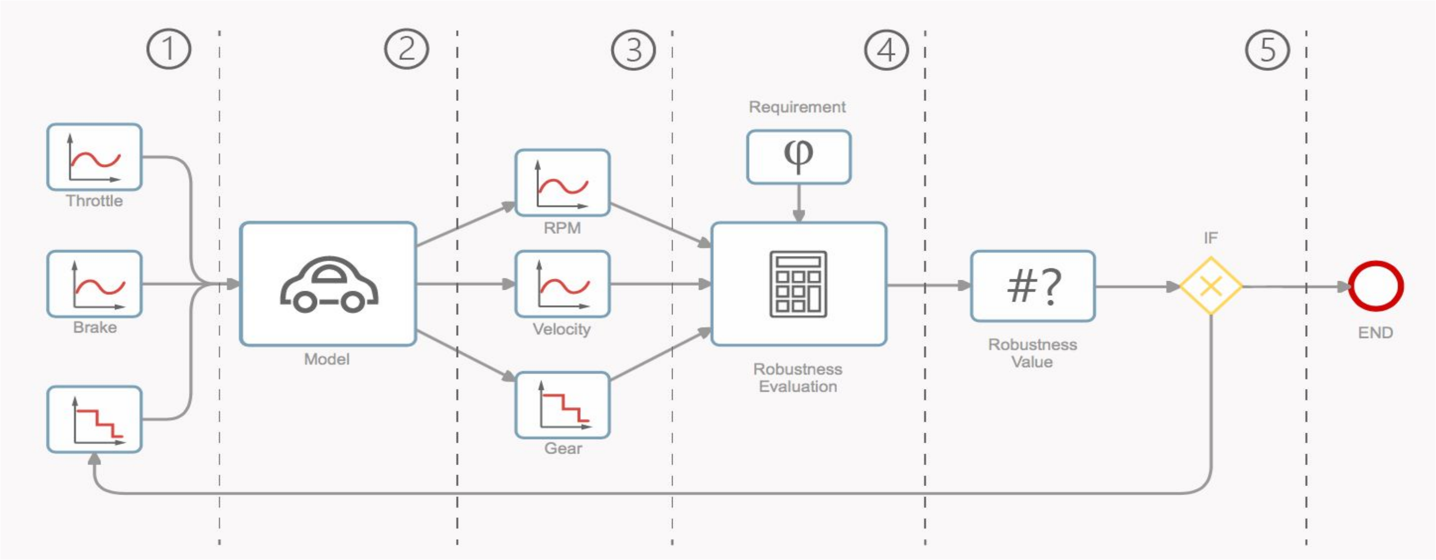
The optimization process



Challenges

- Low number of model execution
- Inputs are functions (temporal series)!!

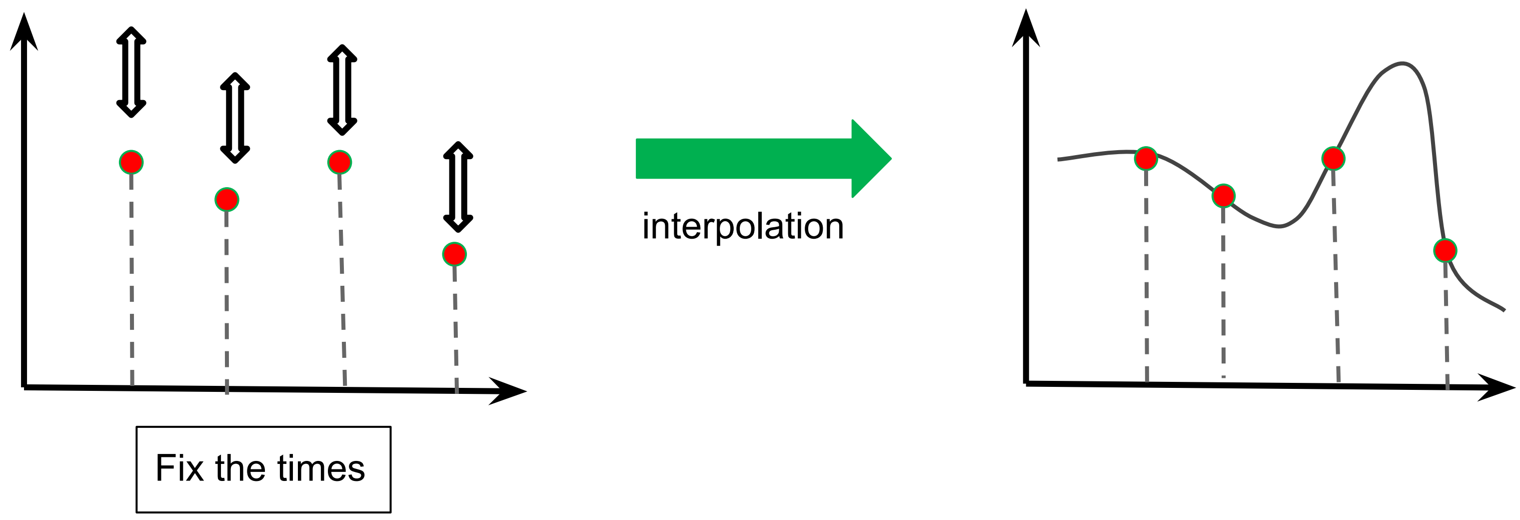
The optimization process



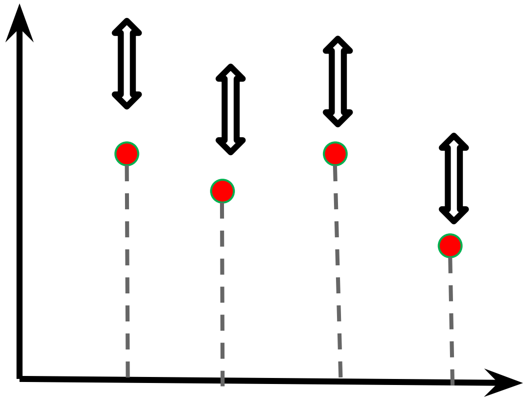
Challenges

- Low number of model execution → GP-UCB
- Inputs are functions (temporal series)!! → Adaptive Control Point Parametrization

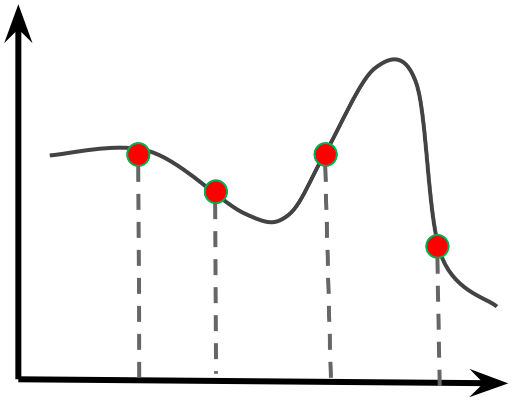
The Control Point Parametrization



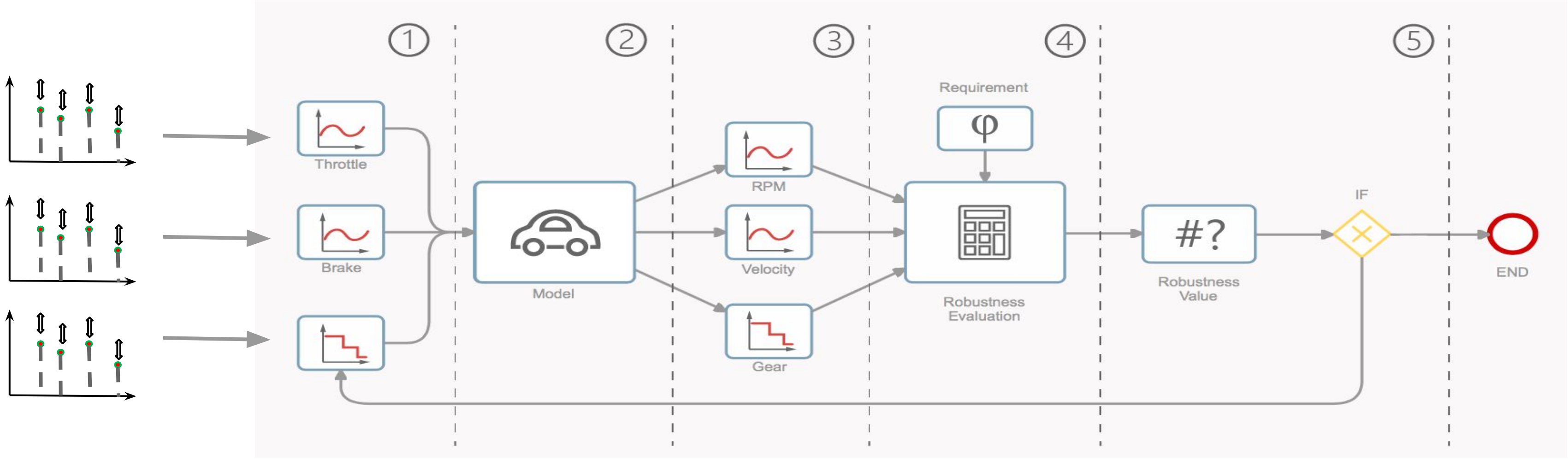
The Control Point Parametrization



interpolation

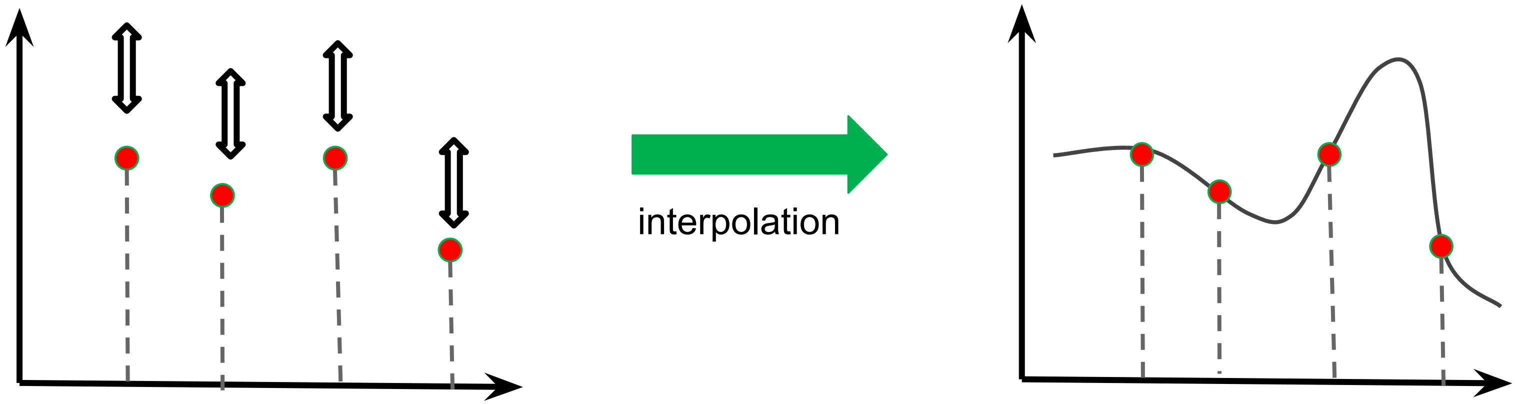


Fix the times

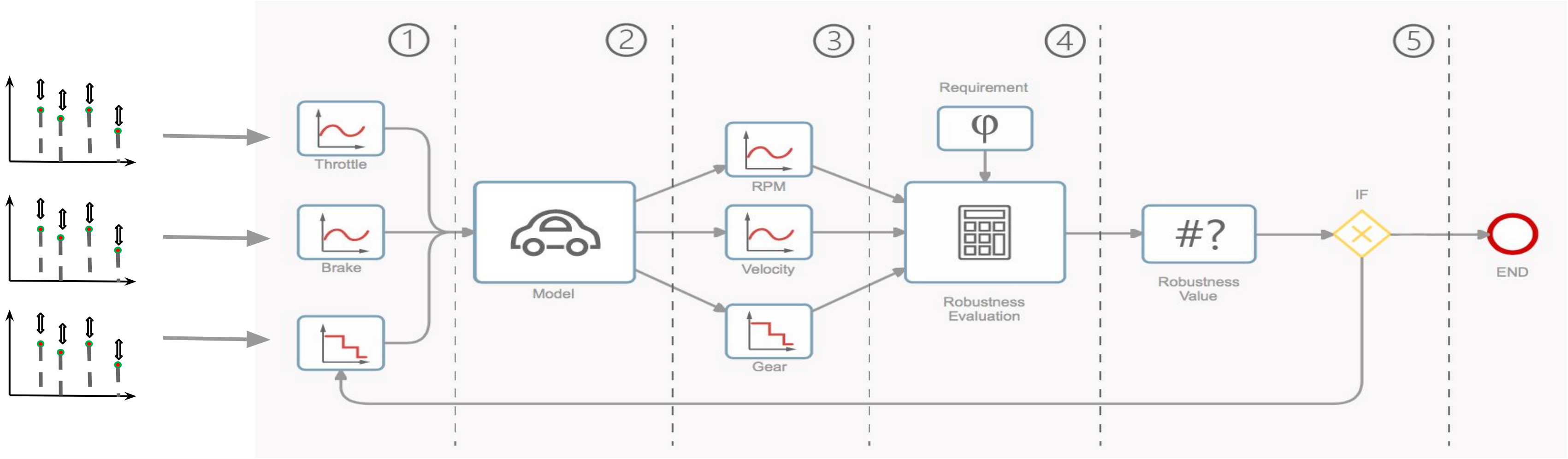


n Control Points → n Variable to optimize

The Control Point Parametrization

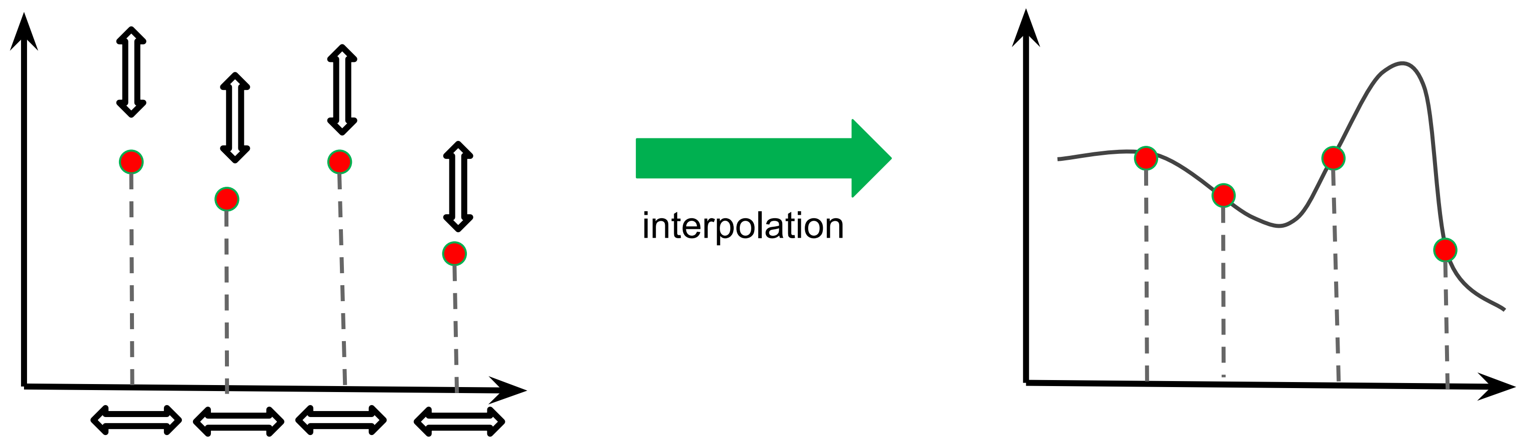


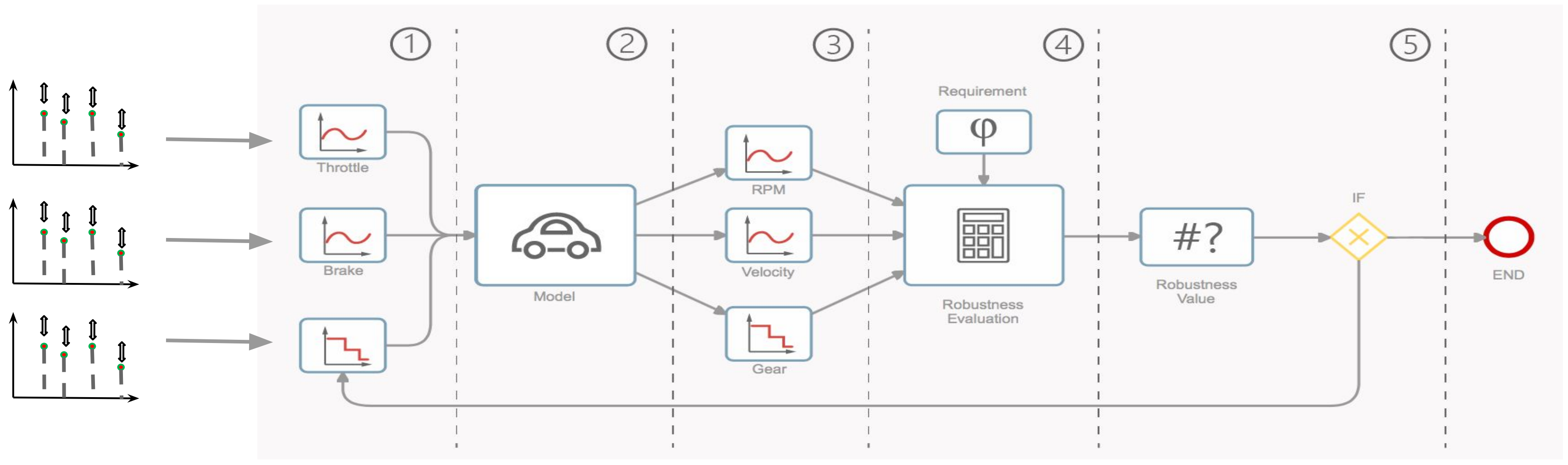
Fix the times



n Control Points → n Variable to optimize

The adaptive Control Point Param.





n Control Points → **2n** Variable to optimize

Problem

Increase the expressivity

but...

Doubled the variables

Problem

Increase the expressivity

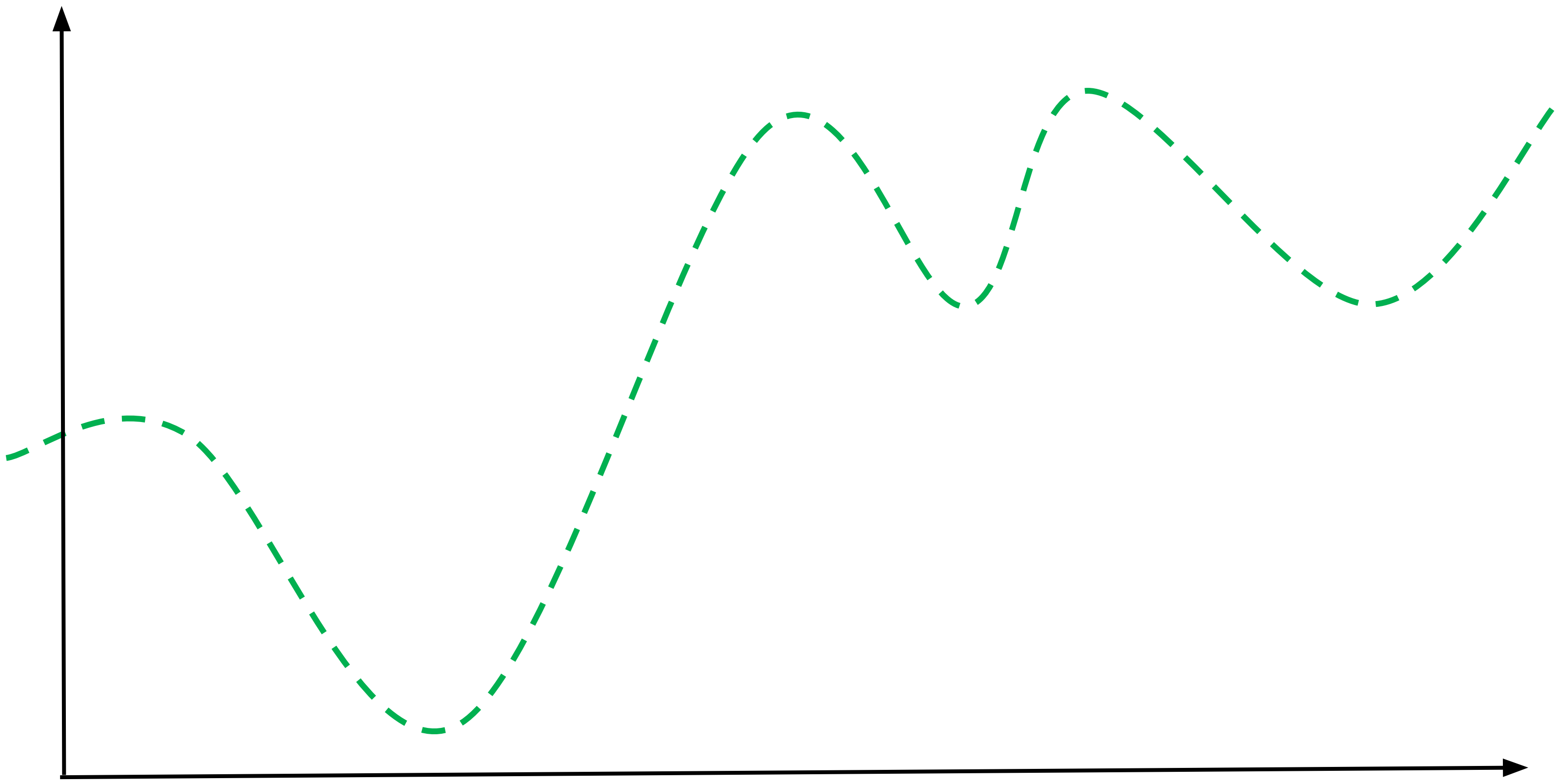
but...

Doubled the variables

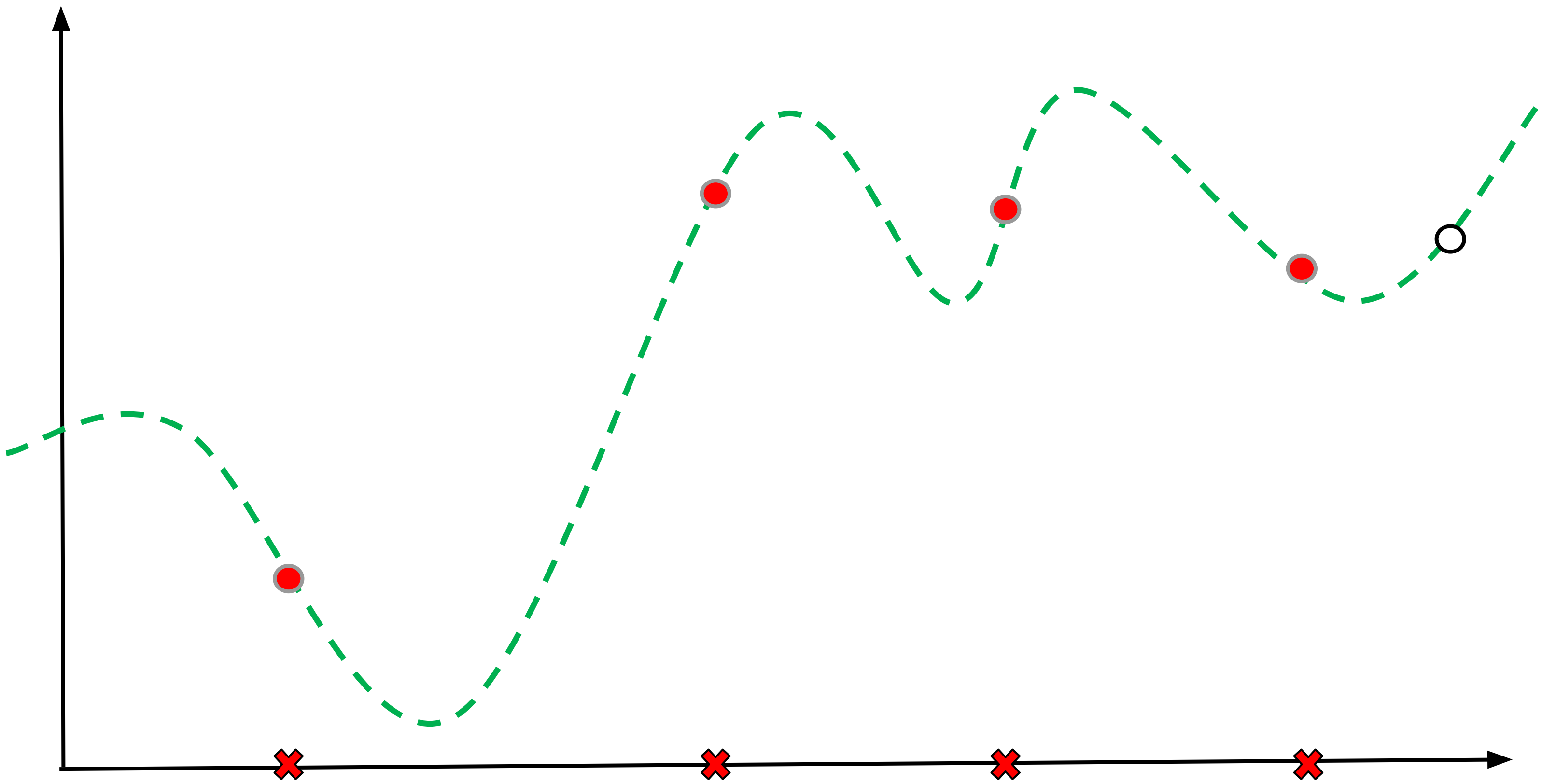
Solution

GP-UCB Optimizer

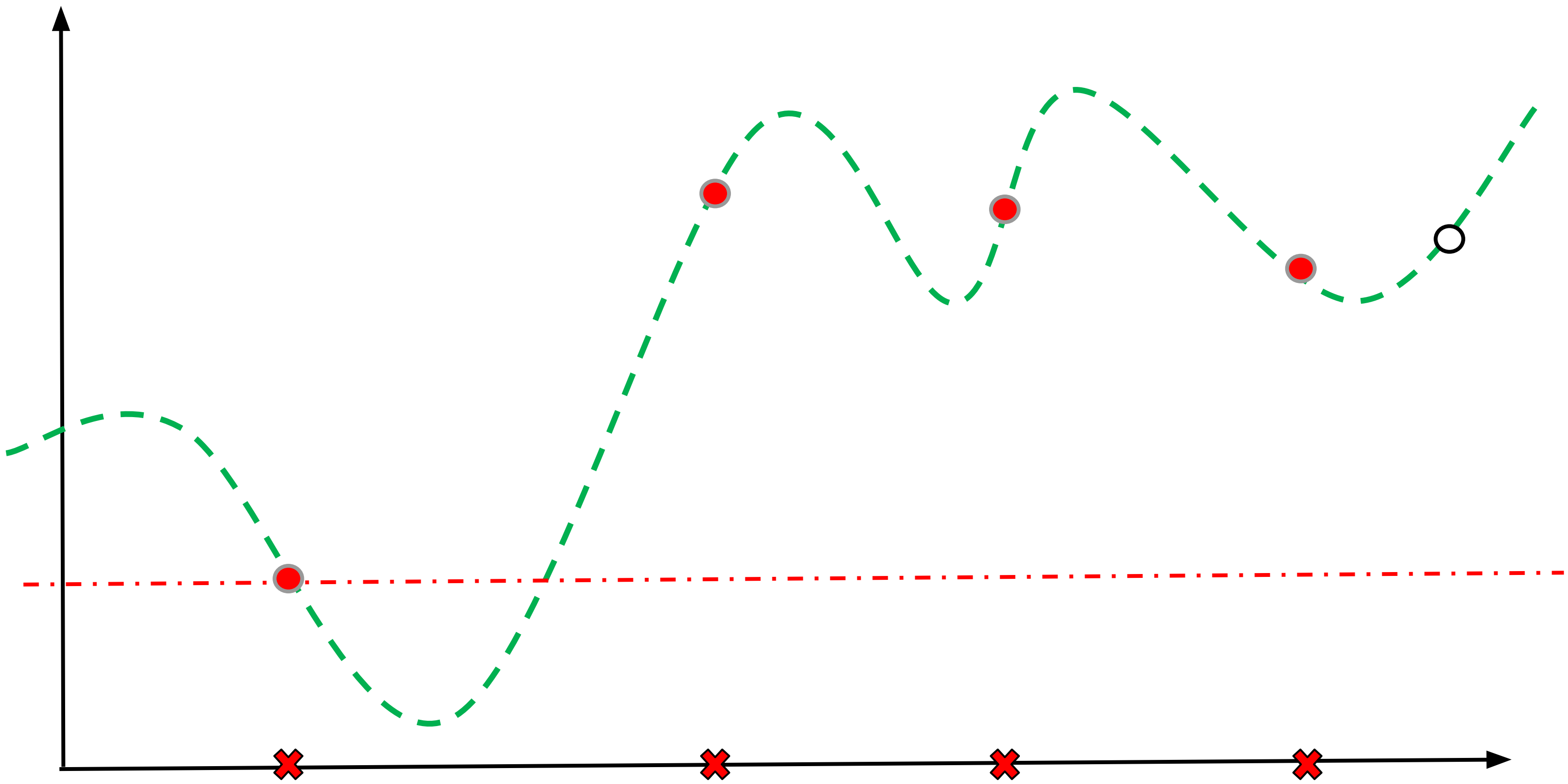
GP-UCB



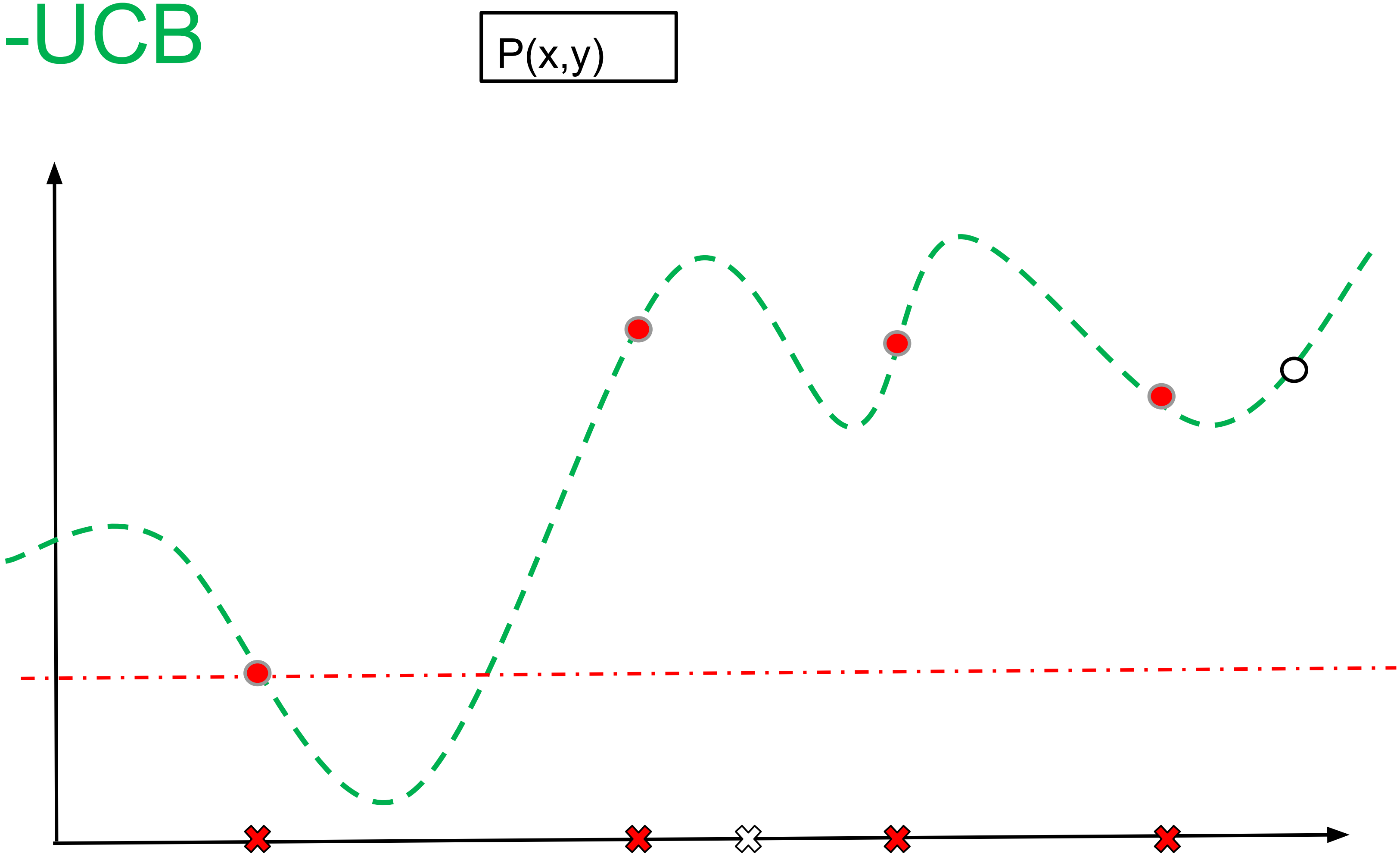
GP-UCB



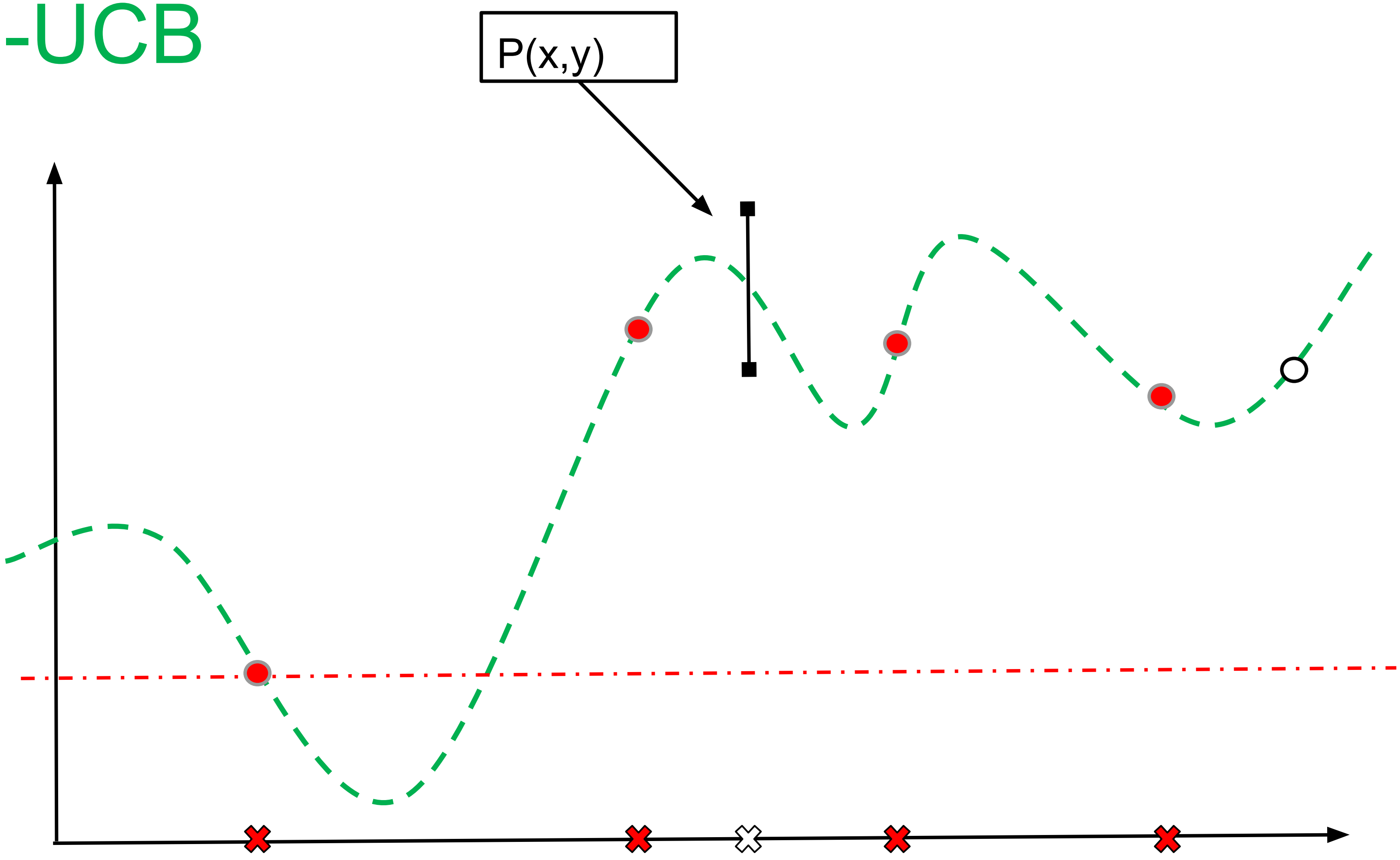
GP-UCB



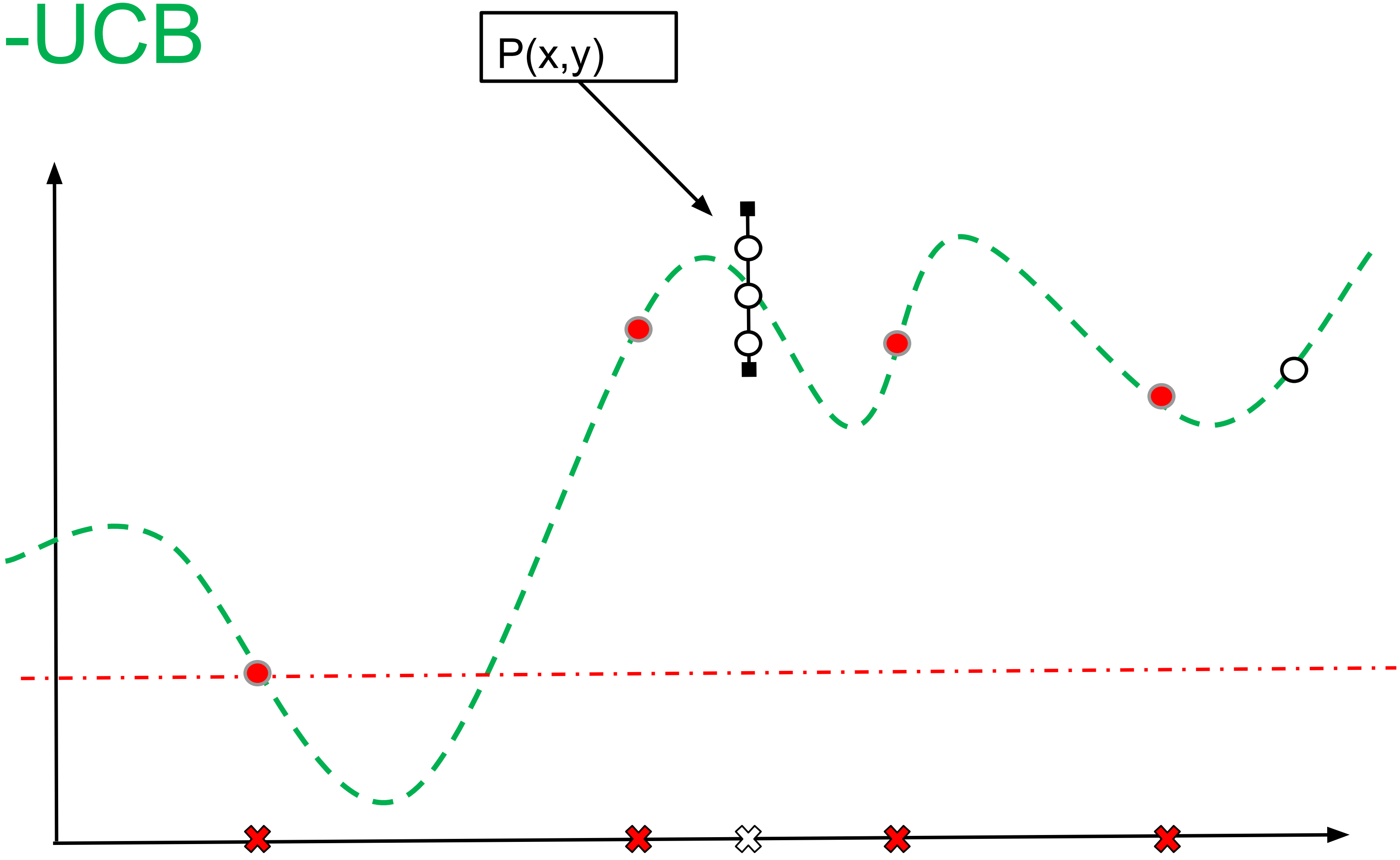
GP-UCB



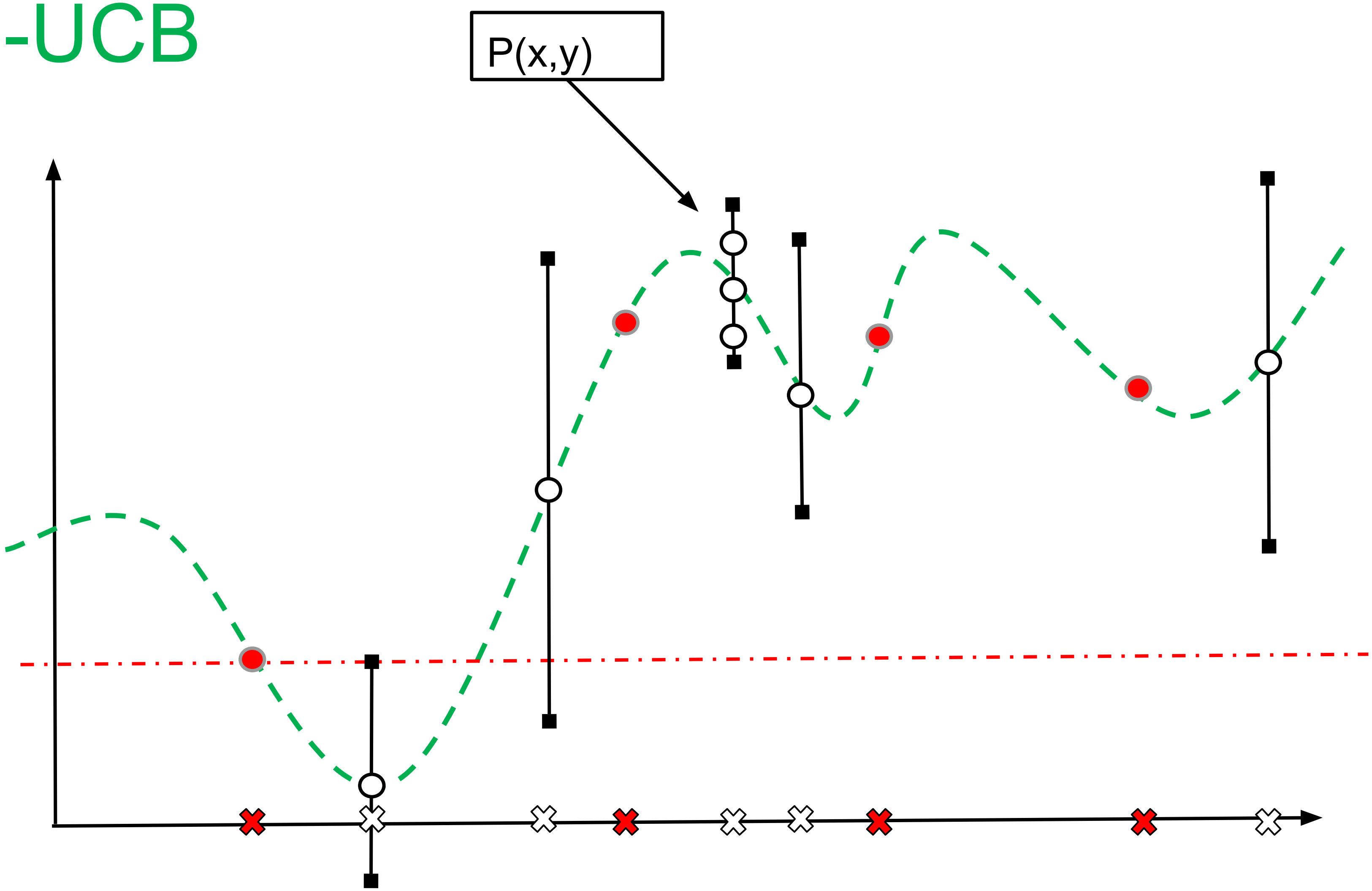
GP-UCB



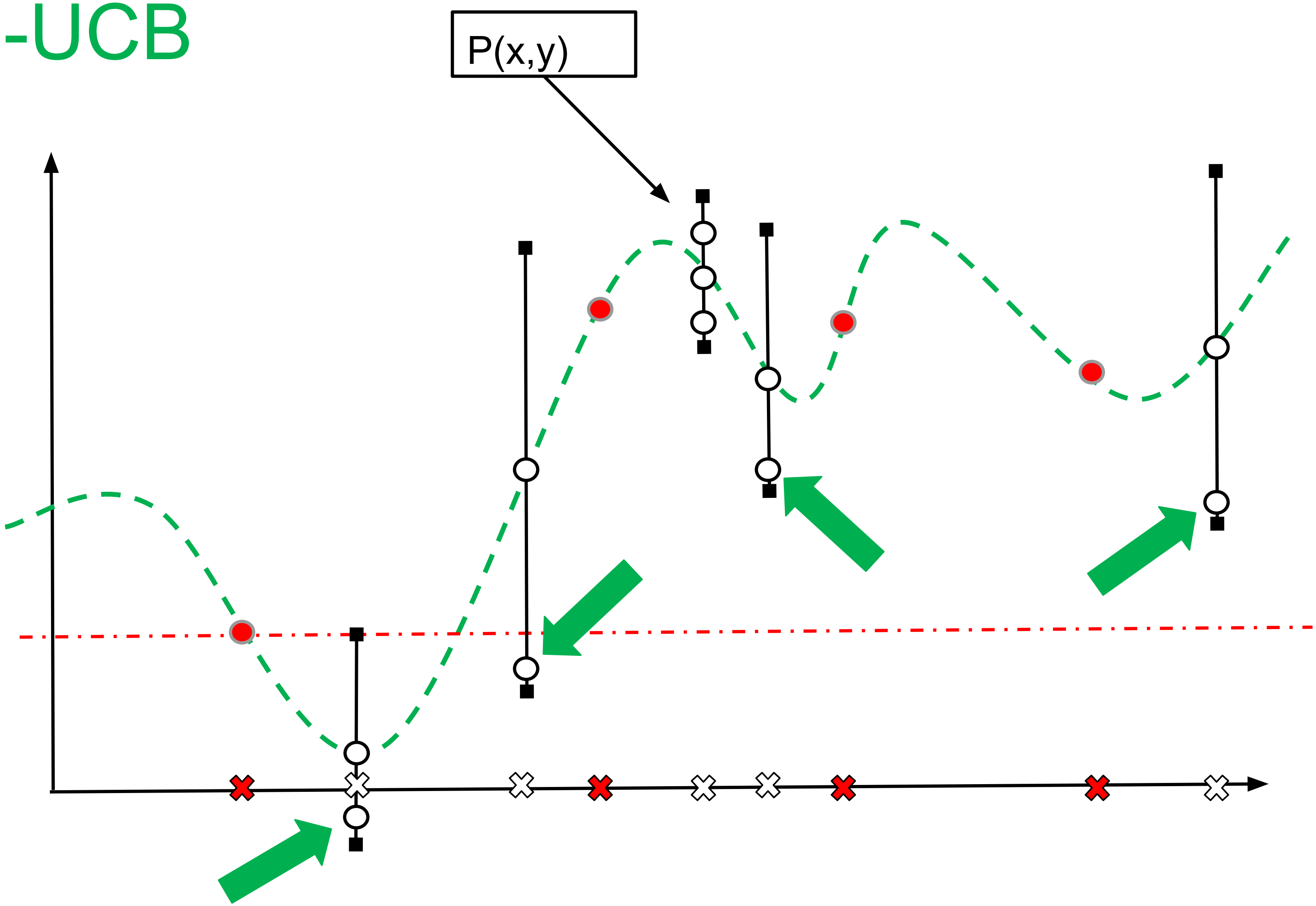
GP-UCB



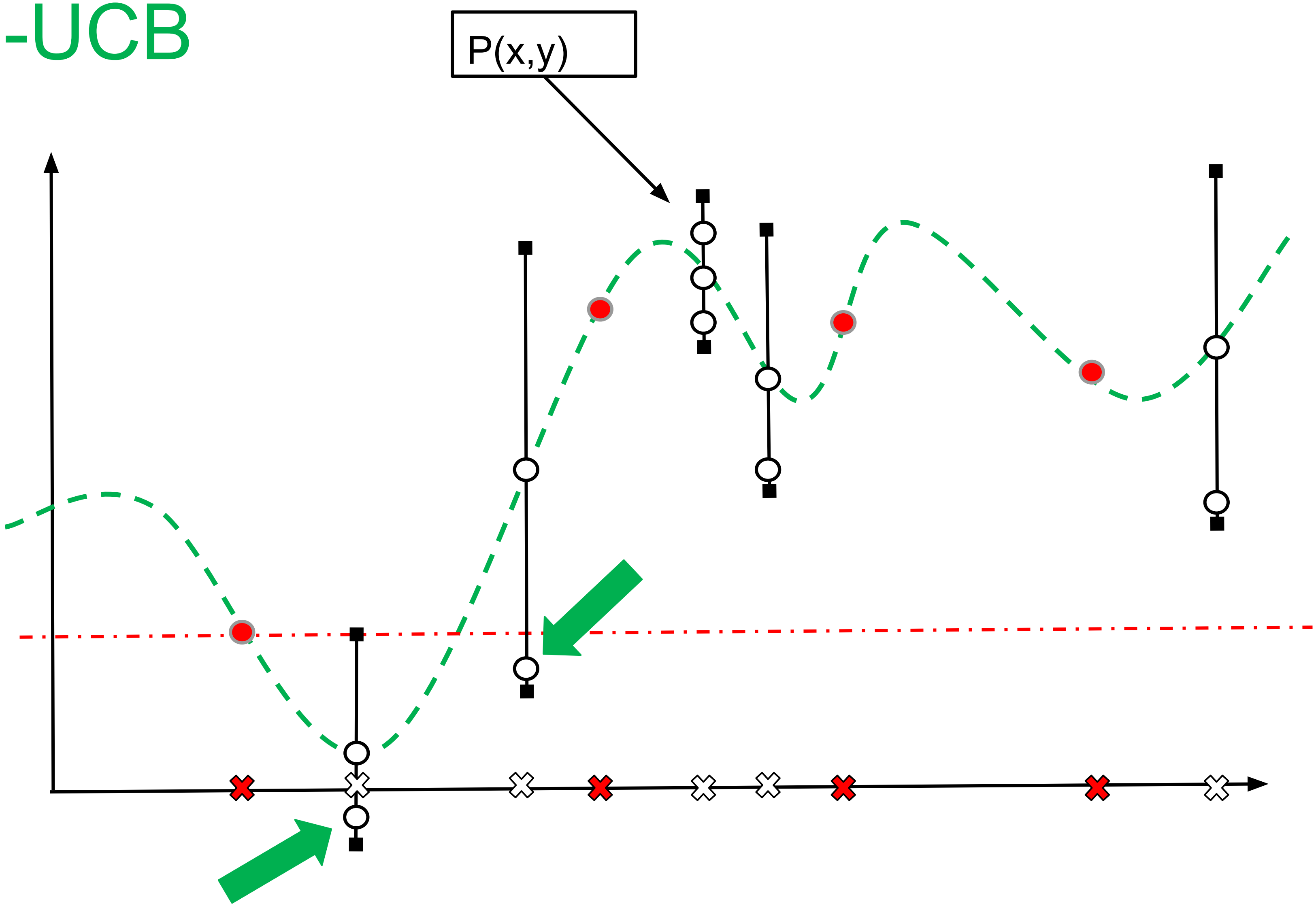
GP-UCB



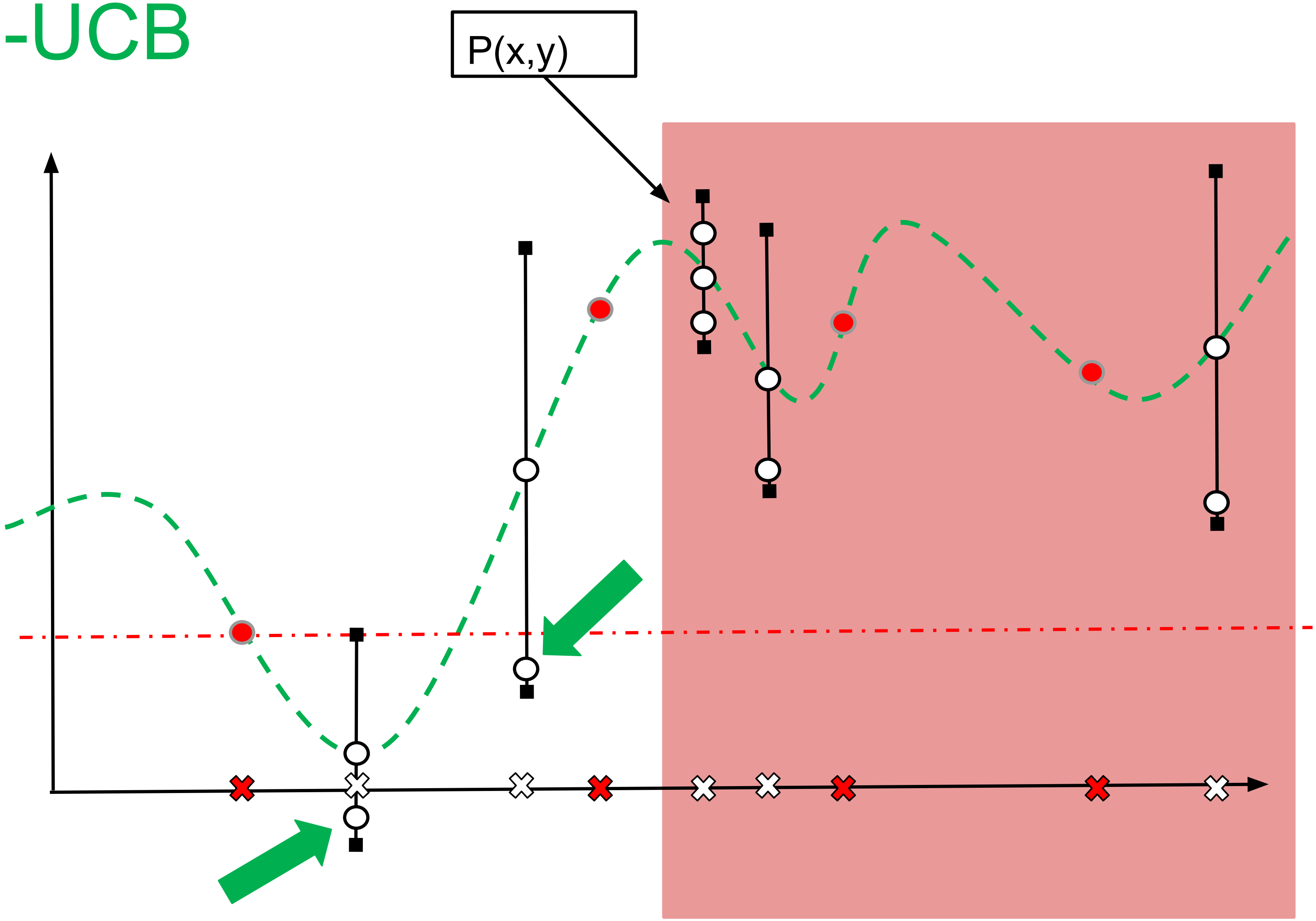
GP-UCB



GP-UCB

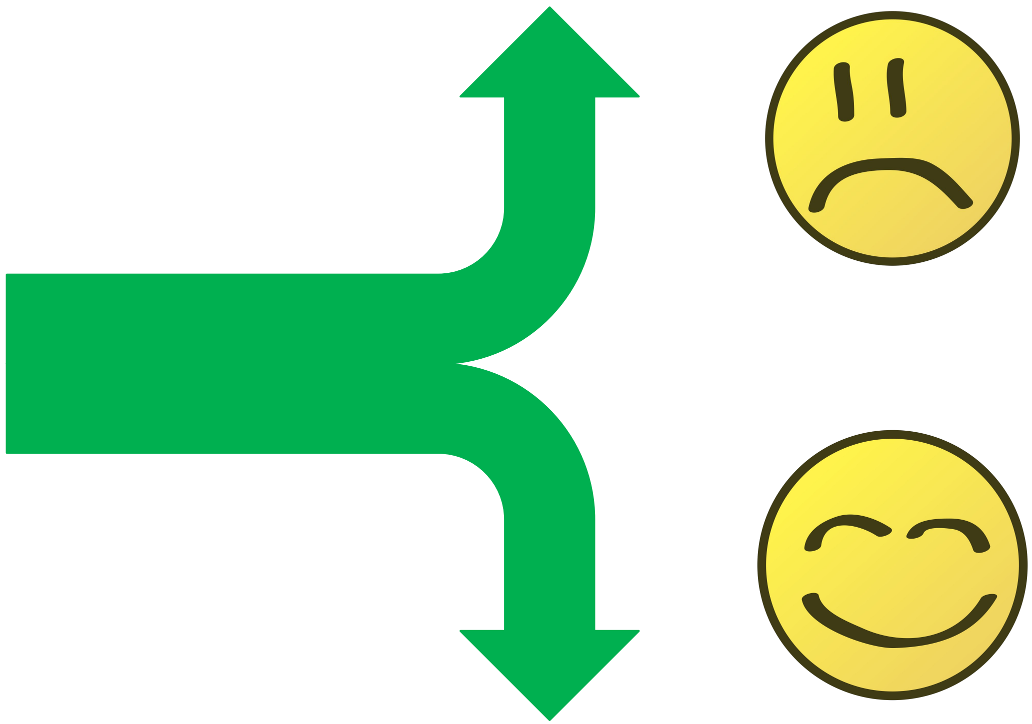


GP-UCB

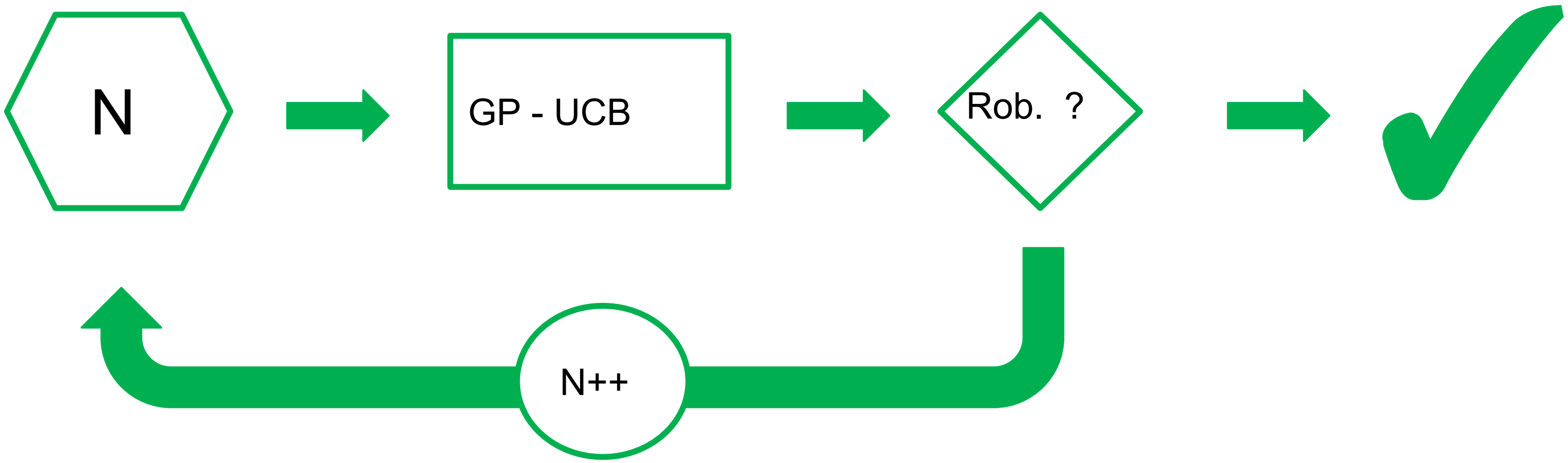


Doubled the variables

Reduce Input Space



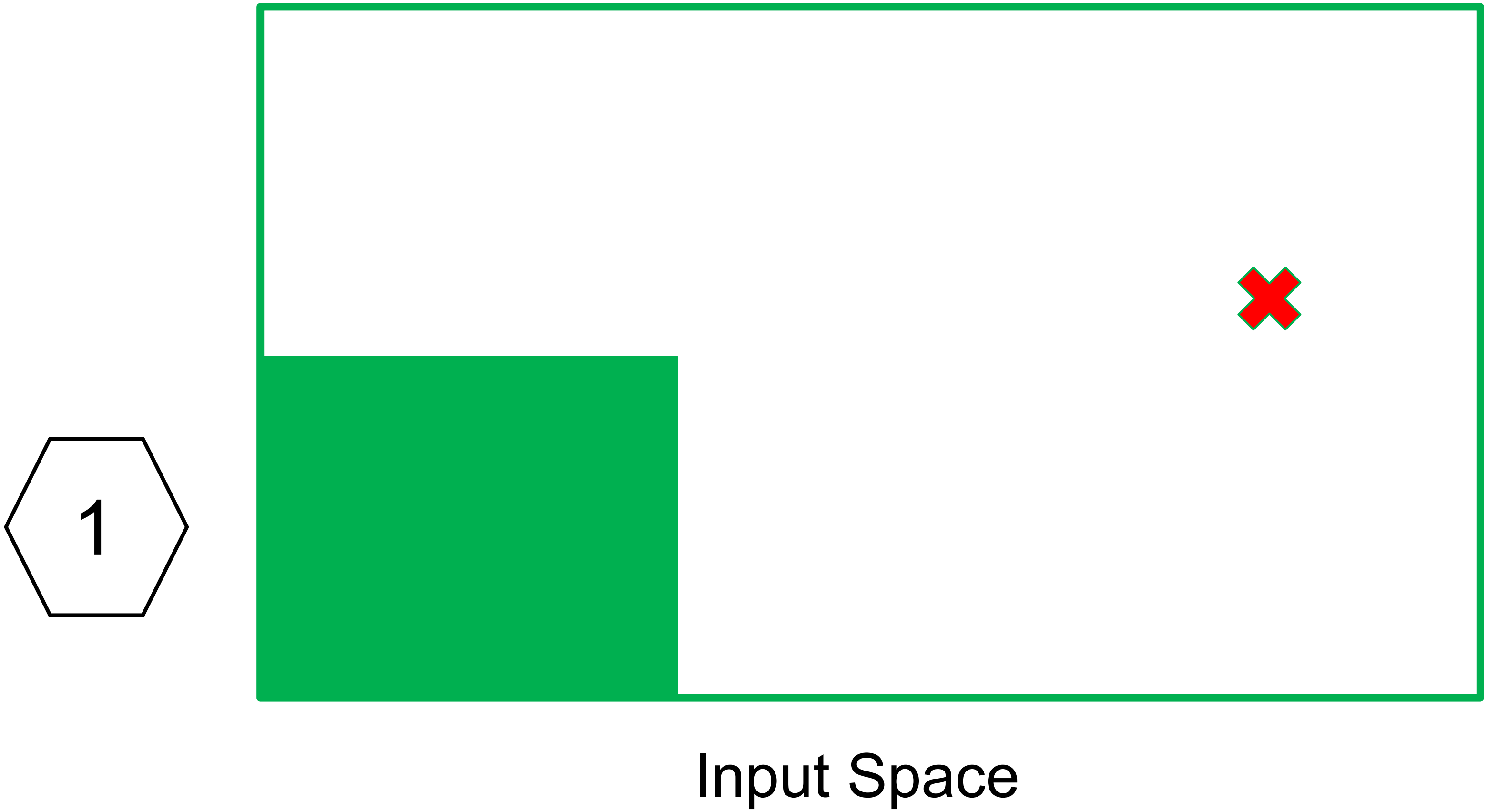
Schema



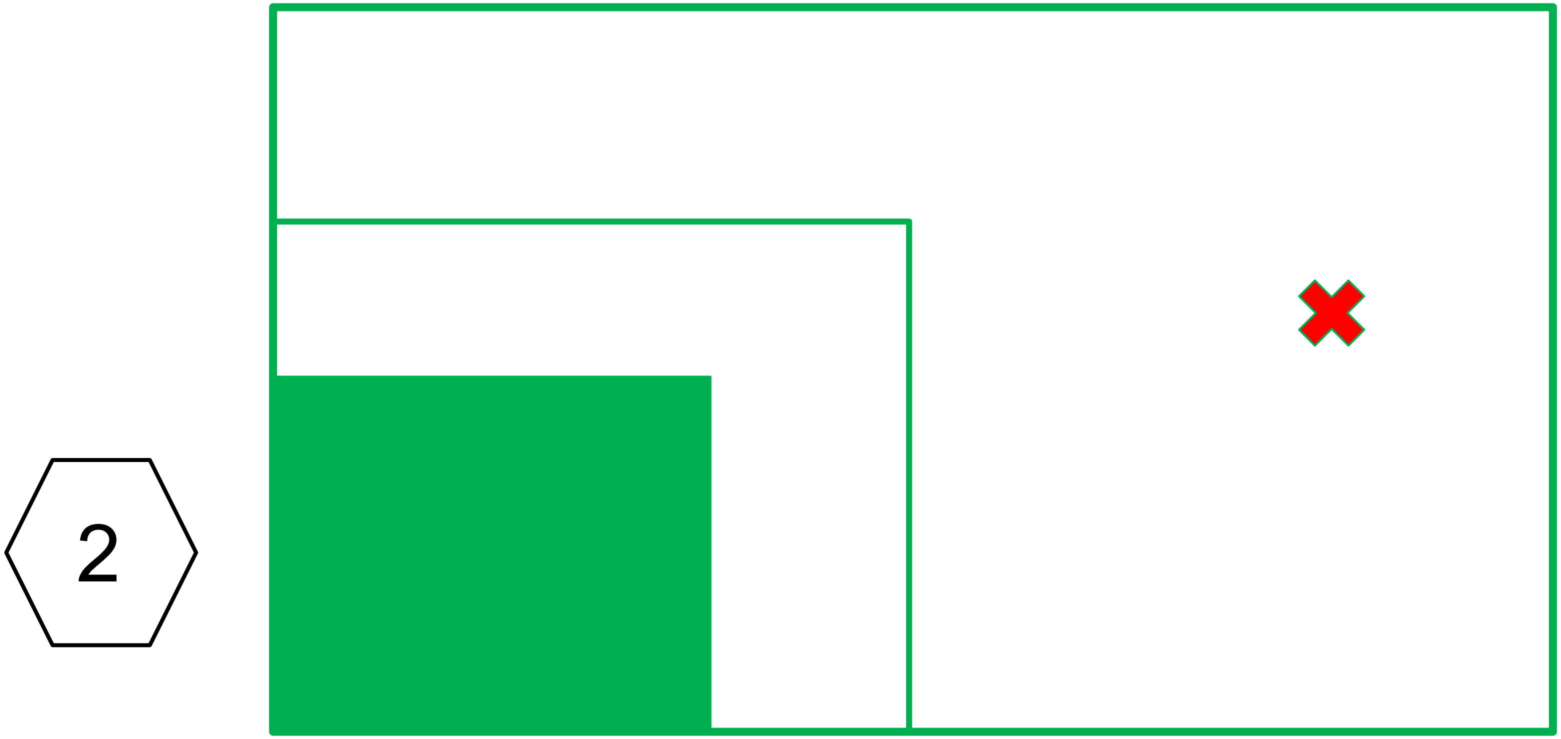
Adaptive Idea

Input Space

Adaptive Idea

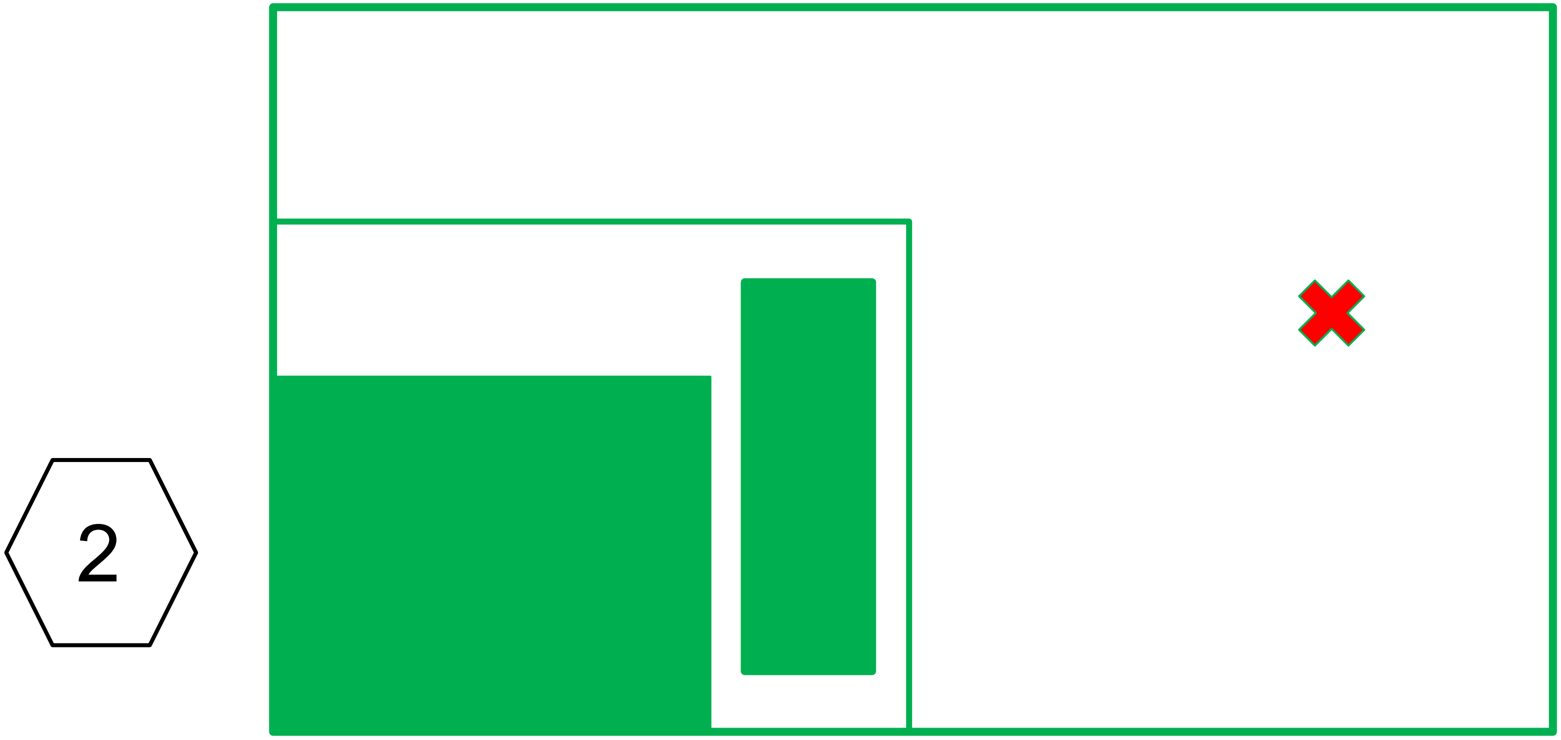


Adaptive Idea



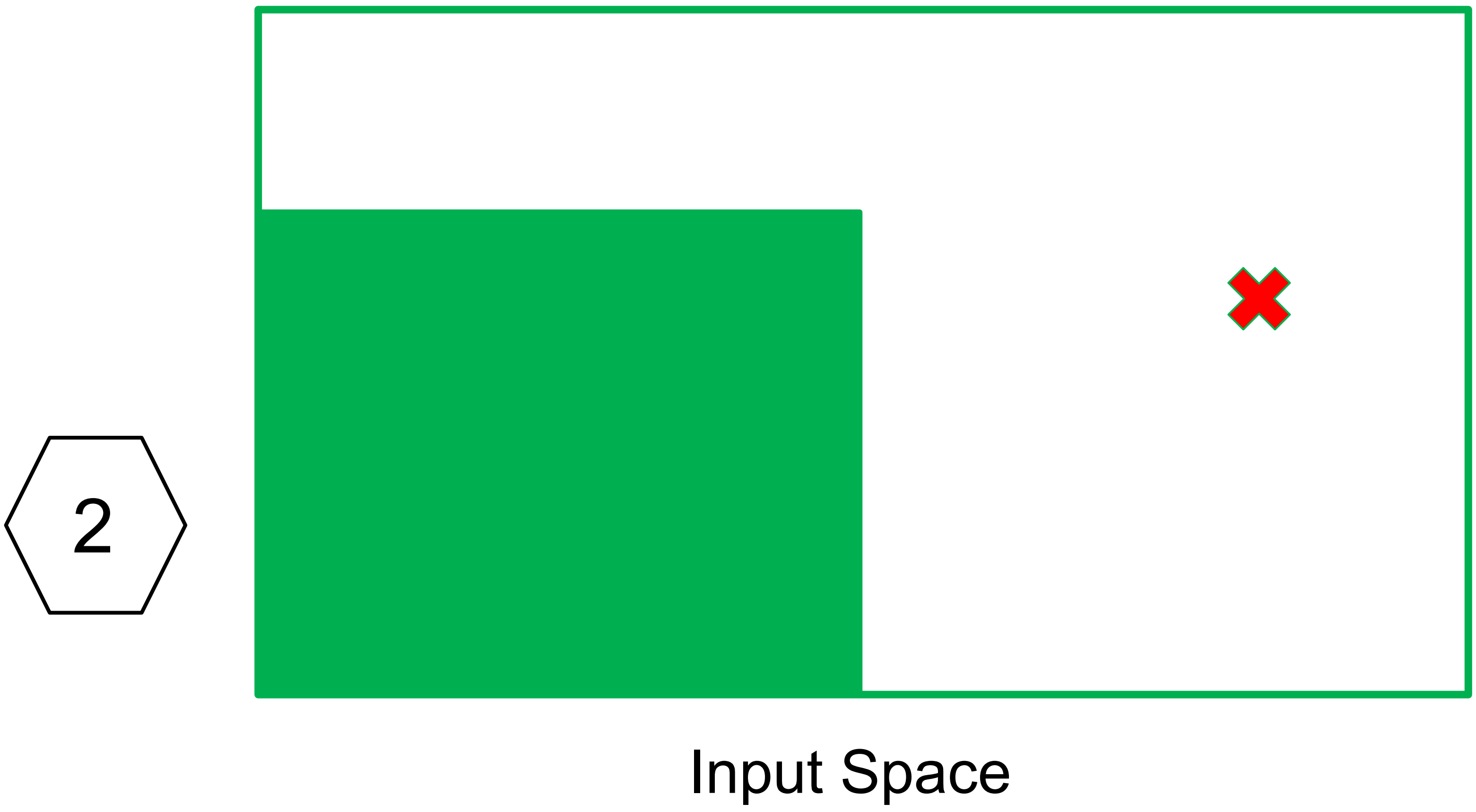
Input Space

Adaptive Idea



Input Space

Adaptive Idea



Adaptive Idea

Input Space

Adaptive Idea

Input Space

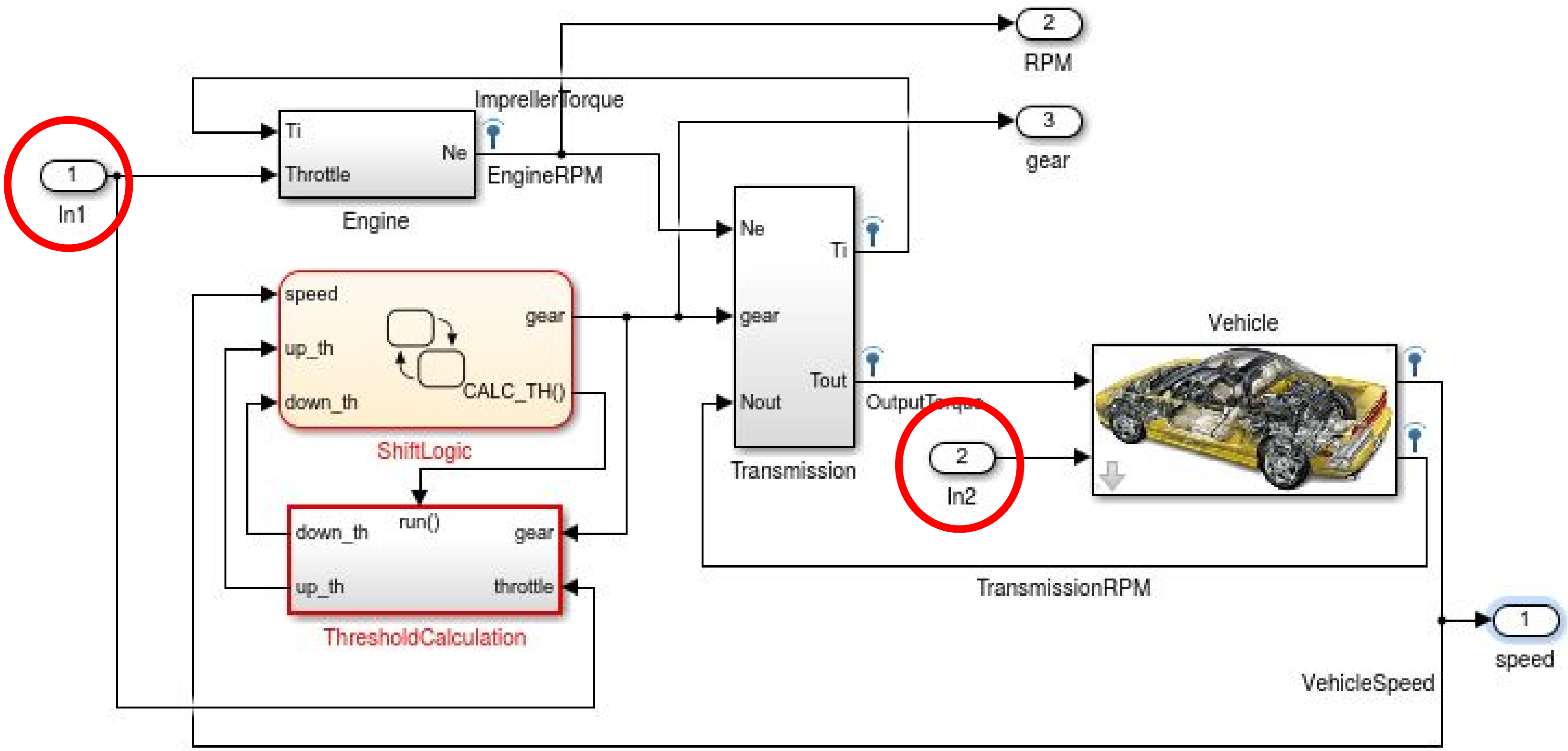
Adaptive Idea

Input Space

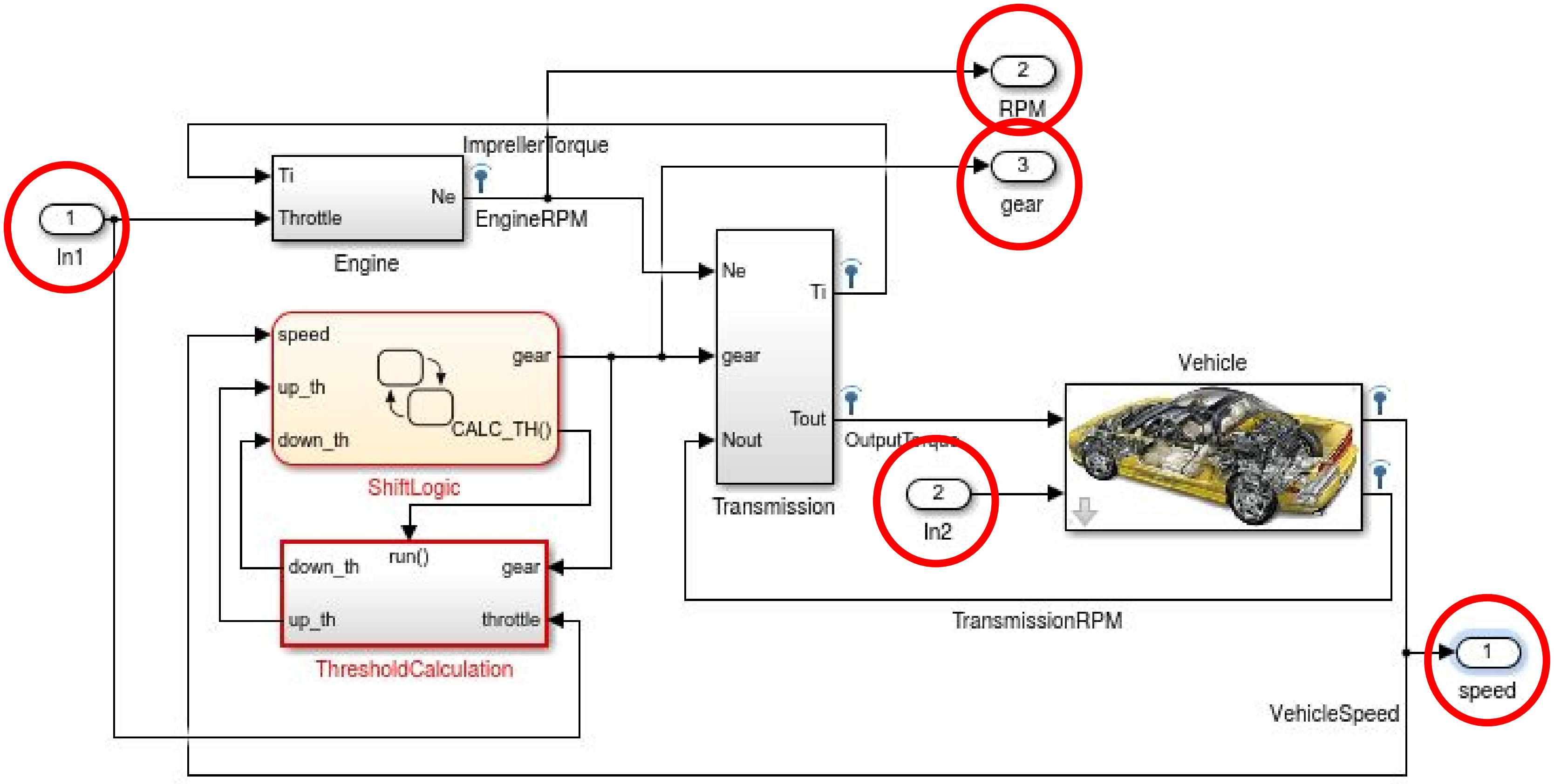
Adaptive Idea

Input Space

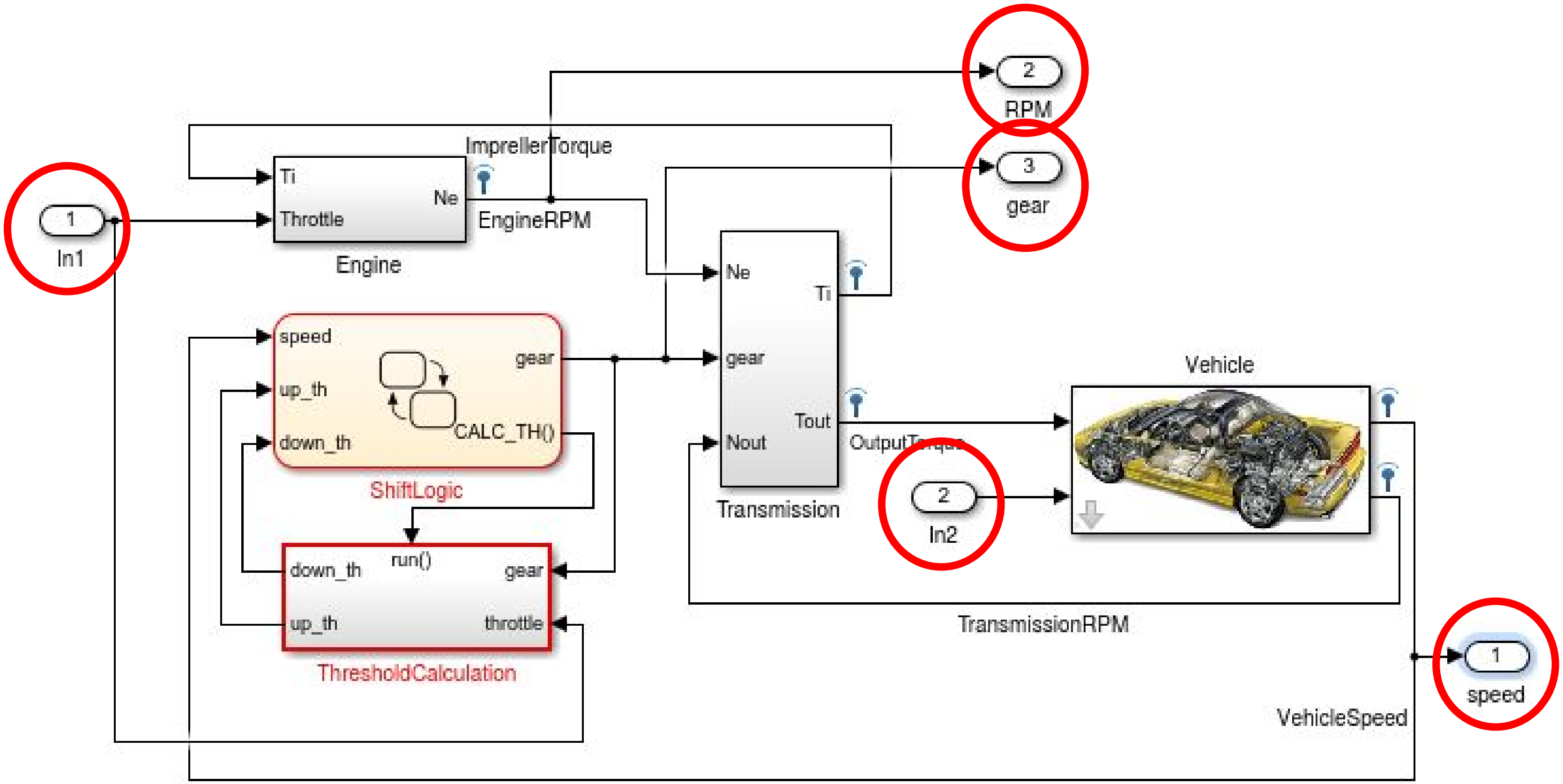
Automatic transmission



Automatic transmission



Automatic transmission



69 blocks: 2 integrators, 3 look-up tables, 3 2D look-up tables, Stateflow Chart

Results

Automatic Transmission		
	Natural languages	**MTL**
φ_1	The engine (w) and the vehicle speed (v) never reach k_1 and k_2, resp.	$\mathbf{G}((w \leq k_1) \wedge (v \leq k_2))$
φ_2	If the engine speed (w) is always less than k_1 then vehicle speed (v) can not exceed k_2 in less then T sec.	$\neg(\mathbf{F}_{[0,T]}(v \geq k_2) \wedge \mathbf{G}(w \leq k_1))$
φ_3	Within T sec the vehicle speed (v) is above k_2 and from that point on the engine speed (w) is always less then k_1	$\mathbf{F}_{[0,T]}((v \geq k_2) \wedge \mathbf{G}((w \leq k_1))$
φ_4	A gear increase from first to fourth in under than 10 sec, ending in an engine speed (w) above k_1 within 2 sec of that, should result in a vehicle speed (v) above k_2.	$((g_1 \mathbf{U} g_2 \mathbf{U} g_3 \mathbf{U} g_4) \wedge \mathbf{F}_{[0,10]}(g_4 \wedge \mathbf{F}_{[0,2]}(w \geq k_1))) \rightarrow \mathbf{G}_{[0,10]}(g_4 \rightarrow \mathbf{X}(g_4 \mathbf{U}_{[0,1]}(v \geq k_2)))$

Results

φ	S-TaLiro			aCPP
	1,1	3,3	7,7	
φ_1 (k_1=4500 , k_2=160)	8,54 ± 5,72	10 ± 10,1	11,69 ± 17,63	4,40 ± 1,01
φ_2 (k_1=4500 , k_2=85)	63,90 ± 53,20	124,82 ± 101,51	1686,9 ± 589,57	18,2 ± 3,5
φ_3 (k_1=4500 , k_2=80)	12,95 ± 7,37	49,8 ± 55,47	199,15± 175,94	6,70 ± 1,74
φ_4 (k_1=4500 , k_2=80)	28,59 ± 24,15	32,65 ± 27,05	26,72 ± 23,98	5,33 ± 1,31

✓ aCPP reduces minimum number of evaluations by **50-70%**

✗ GP-UCB is slow.

Results

Time = {#Simulations} x {Simulation Time} + {Optimizer time}

GP-UCB is slow

Results

Time = {#Simulations} x {Simulation Time} + {Optimizer time}

GP-UCB is slow

Future work

- from **Matlab** to **Java** (parallelization)
- multi-objective approach
- using fmi as simulator

Acknowledges

Esteco

Luca Bortolussi

Alberto Policriti

Thank You!

....and use Formal Methods