Influence of uncertainties on ultrasonic localization systems

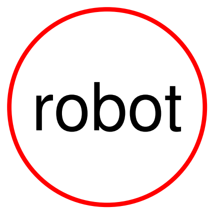
J-P. Merlet
COPRIN Project team

INRIA

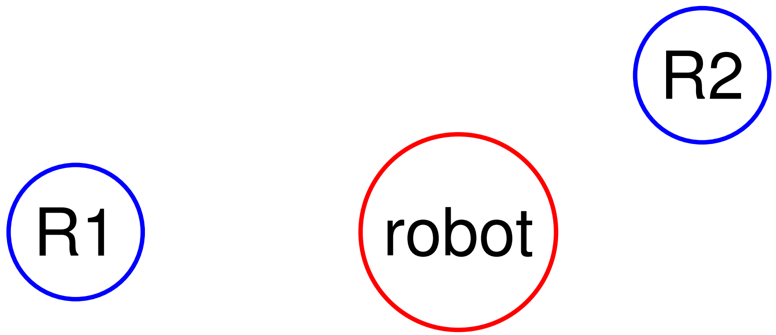
Joint work with ETH Zurich

Problem: localizing a robot which has a sound-emitter (single ping)

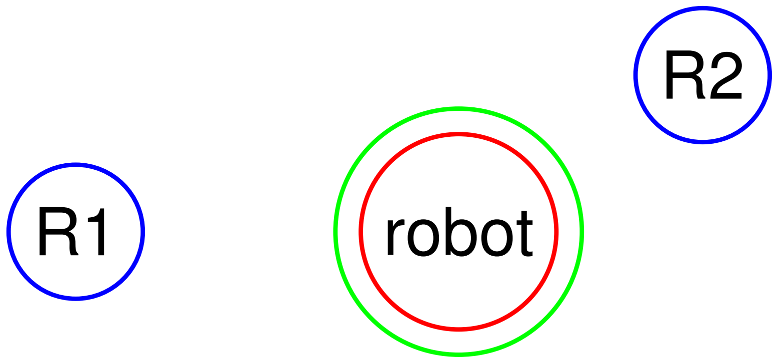
Problem: localizing a robot which has a sound-emitter (single ping)



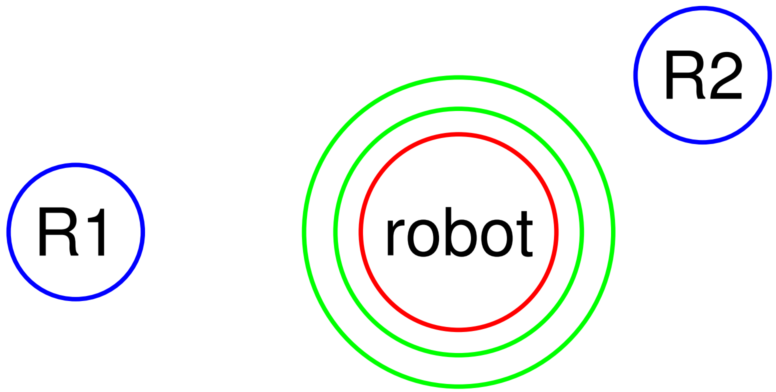
Problem: localizing a robot which has a sound-emitter (single ping)



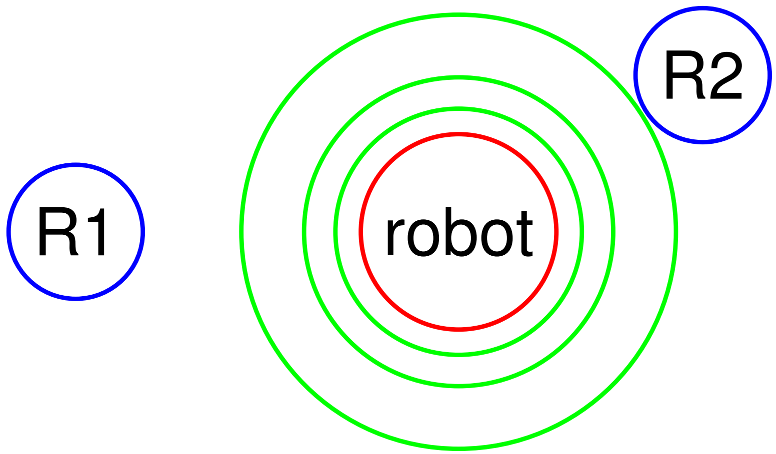
Problem: localizing a robot which has a sound-emitter (single ping)



Problem: localizing a robot which has a sound-emitter (single ping)

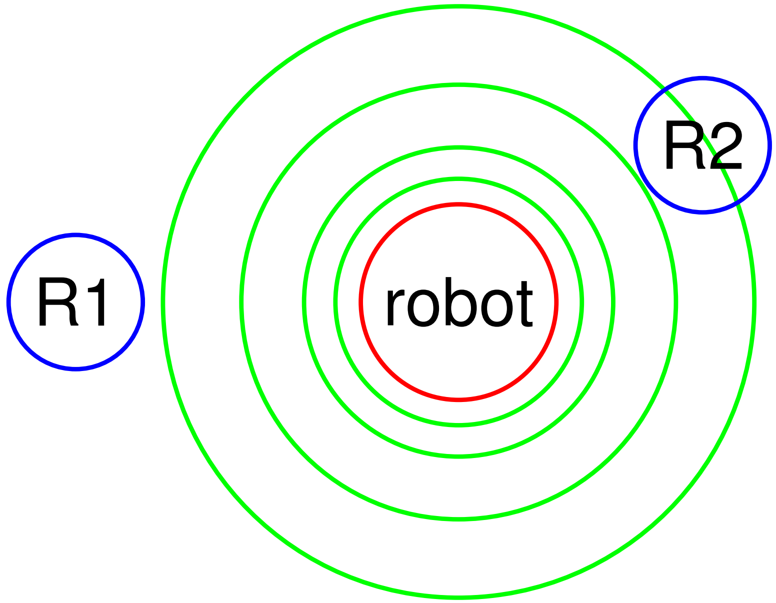


Problem: localizing a robot which has a sound-emitter (single ping)

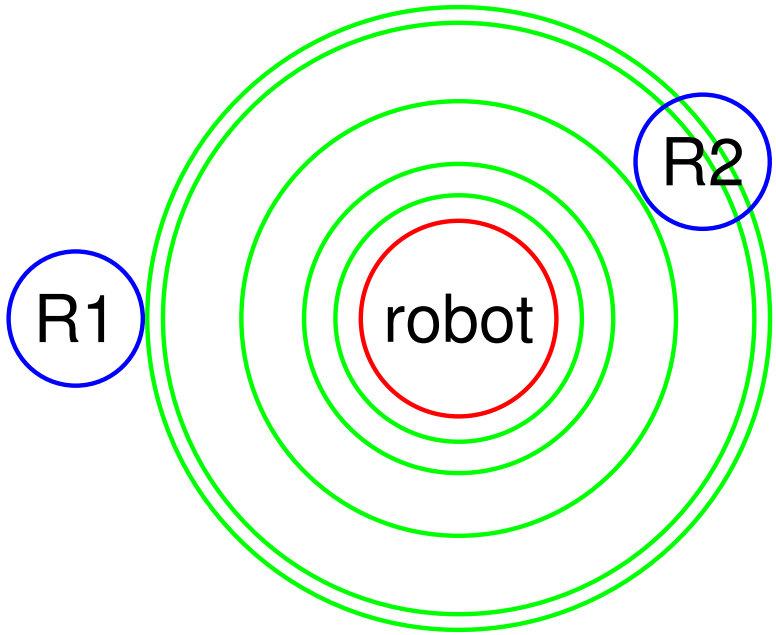


R2 receives the sound signal at time t_2

Problem: localizing a robot which has a sound-emitter (single ping)



Problem: localizing a robot which has a sound-emitter (single ping)



R1 receives the sound signal at time t_1

Time Difference Of Arrival (TDOA): $t_1 - t_2$

Time Difference Of Arrival (TDOA): $t_1 - t_2$

- d_i: distance between robot and receiver R_i
- c: sound velocity

Time Difference Of Arrival (TDOA): $t_1 - t_2$

- d_i: distance between robot and receiver R_i
- c: sound velocity

$$TDOA = \frac{d_1 - d_2}{c}$$

Time Difference Of Arrival (TDOA): $t_1 - t_2$

- d_i: distance between robot and receiver R_i
- c: sound velocity

$$TDOA = \frac{d_1 - d_2}{c}$$

TDOA is measured, robot location such that $d_1 - d_2$ has a fixed value

Time Difference Of Arrival (TDOA): $t_1 - t_2$

- d_i: distance between robot and receiver R_i
- c: sound velocity

$$TDOA = \frac{d_1 - d_2}{c}$$

TDOA is measured, robot location such that $d_1 - d_2$ has a fixed value

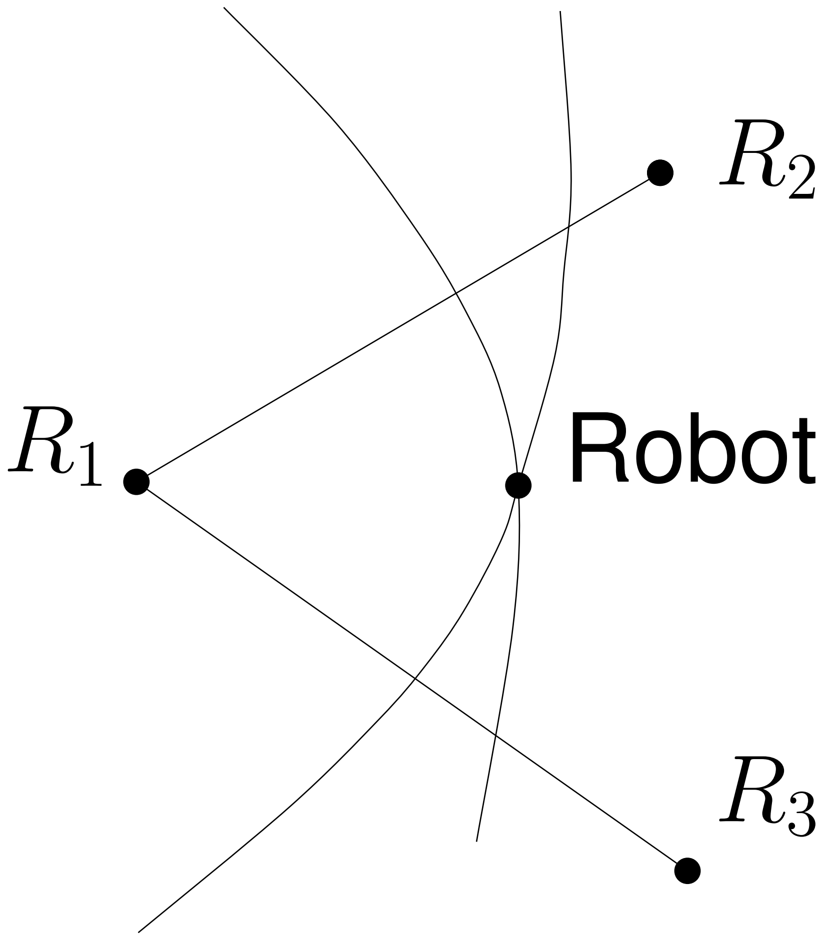
$\Rightarrow$ robot is on a hyperbola having R_1, R_2 as focal points

INRIA SOPHIA ANTIPOLIS

3 receivers: robot is at the intersection of 2 hyperbola

$\Downarrow$

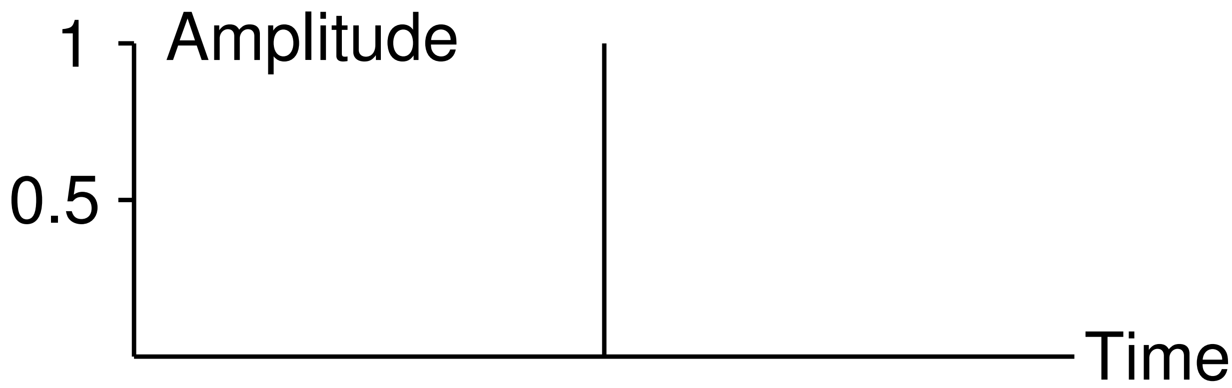
location is, in general, unique



The practice

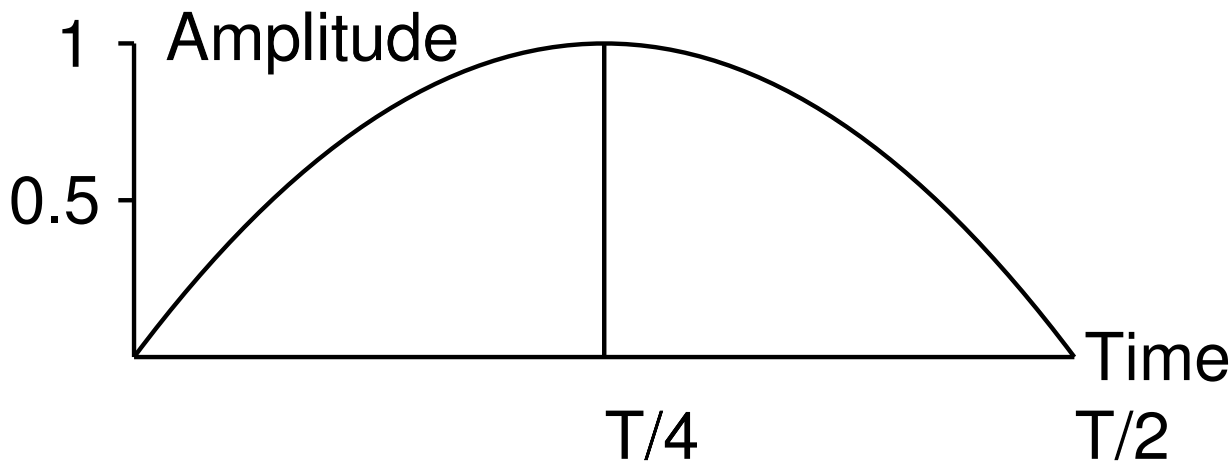
The practice

- the sound signal is not a ping



The practice

- the sound signal is a distribution with frequency f



The practice

- the sound signal is a distribution with frequency f
- emitter location has a probability distribution

INRIA SOPHIA ANTIPOLIS

The practice

- the sound signal is a distribution with frequency f
- emitter location has a probability distribution
- this probability distribution is dependent upon:

INRIA SOPHIA ANTIPOLIS

The practice

- the sound signal is a distribution with frequency f
- emitter location has a probability distribution
- this probability distribution is dependent upon:
  - sound velocity c

The practice

- the sound signal is a distribution with frequency f
- emitter location has a probability distribution
- this probability distribution is dependent upon:
    - sound velocity c
    - TDOA

The practice

- the sound signal is a distribution with frequency f
- emitter location has a probability distribution
- this probability distribution is dependent upon:
  - sound velocity c
  - TDOA
  - ultrasound frequency f

The probability distribution

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

- compute the TDOA for two emitters

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

- compute the TDOA for two emitters
- R_i emits waves fronts with TDOA as time shift

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

- compute the TDOA for two emitters
- R_i emits waves fronts with TDOA as time shift
- waves front interfere

The probability distribution

Revert the problem: $R_i \rightarrow$ emitters, robot $\rightarrow$ receiver

- compute the TDOA for two emitters
- R_i emits waves fronts with TDOA as time shift
- waves front interfere
- constructive interference occurs at real robot location

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

- compute the TDOA for two emitters
- R_i emits waves fronts with TDOA as time shift
- waves front interfere
- constructive interference occurs at real robot location

$$\Downarrow$$

larger signal

INRIA SOPHIA ANTIPOLIS

The probability distribution

Revert the problem: $R_i \longrightarrow$ emitters, robot $\longrightarrow$ receiver

- compute the TDOA for two emitters
- R_i emits waves fronts with TDOA as time shift
- waves front interfere
- constructive interference occurs at real robot location

$$\Downarrow$$

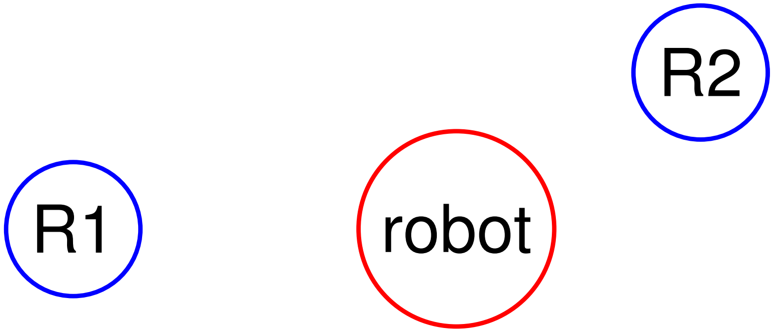
larger signal

- detection if signal is larger than a given threshold

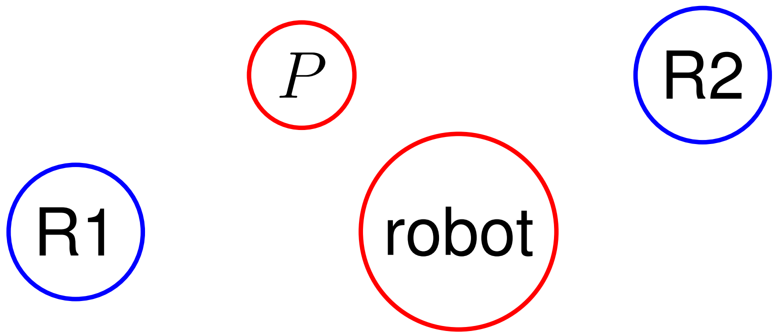
INRIA SOPHIA ANTIPOLIS

Computing the signal

Computing the signal

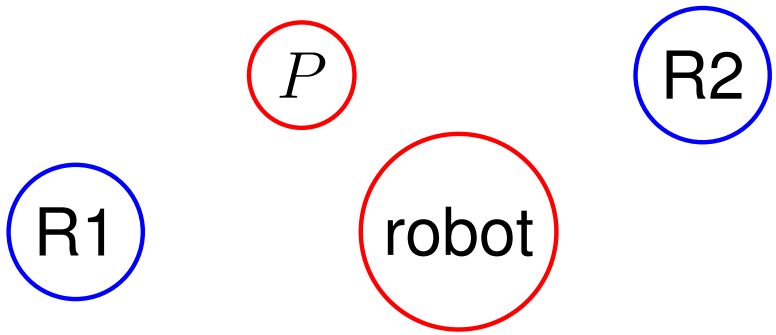


Computing the signal



d'_j: distance between P and R_j

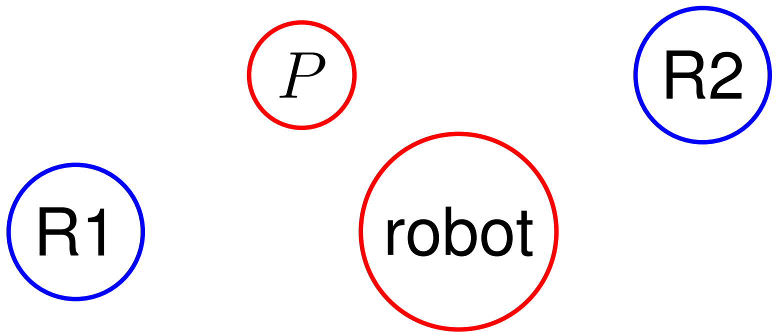
Computing the signal



d'_j: distance between P and R_j

signal amplitude at P for receivers R_i, R_j: A_{ij}

Computing the signal

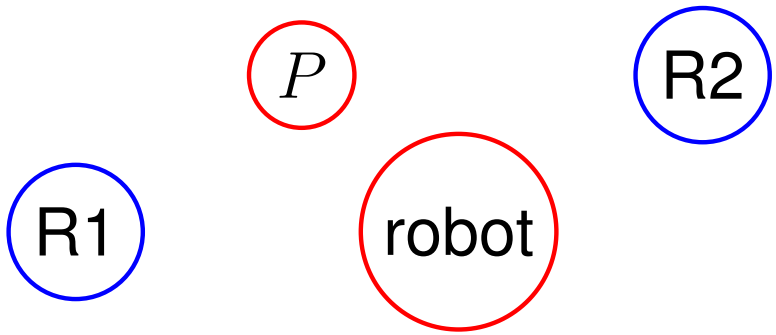


d'_j: distance between P and R_j

signal amplitude at P for receivers R_i, R_j: A_{ij}

$$A_{ij} = \sqrt{2\cos(2\pi \frac{f}{c}(d_i - d'_i - d_j + d'_j)) + 2}$$

Computing the signal



d'_j: distance between P and R_j

signal amplitude at P for receivers R_i, R_j: A_{ij}

$$A_{ij} = \sqrt{2\cos(2\pi\frac{f}{c}(d_i - d'_i - d_j + d'_j)) + 2}$$

maximal signal: 2

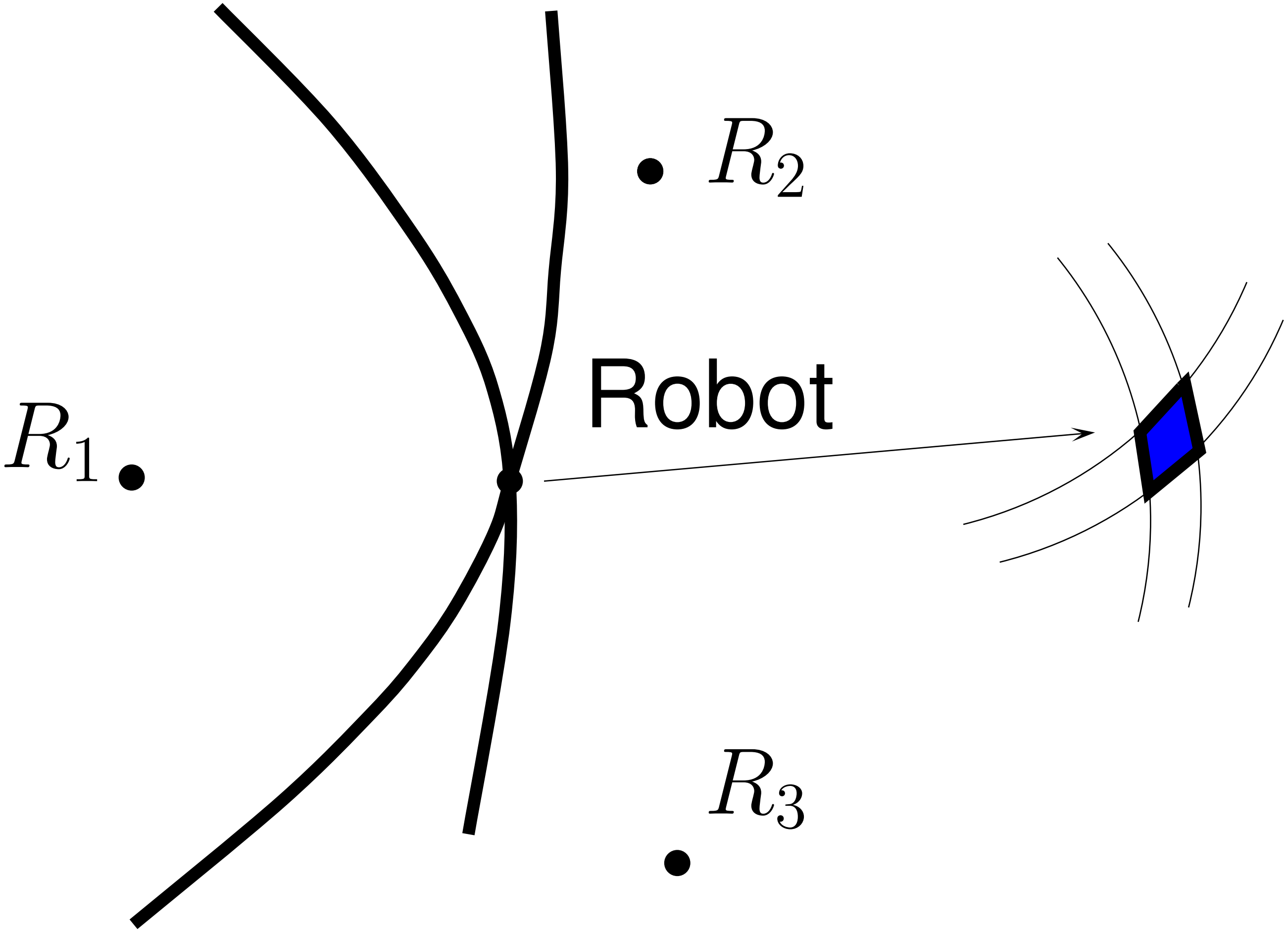
If for all pairs of receiver $A_{ij} >$ threshold: detection

If for all pairs of receiver $A_{ij} >$ threshold: detection

the possible robot location is a region

If for all pairs of receiver $A_{ij} >$ threshold: detection

the possible robot location is a region



Managing uncertainties

Managing uncertainties

- c varies with temperature T

INRIA SOPHIA ANTIPOLIS

Managing uncertainties

- c varies with temperature T
  - in water, $c = 1404.3 + 4.7T - 0.04T^2$

Managing uncertainties

- c varies with temperature T
  - in water, $c = 1404.3 + 4.7T - 0.04T^2$
- f is not known exactly

INRIA SOPHIA ANTIPOLIS

Managing uncertainties

- c varies with temperature T
  - in water, $c = 1404.3 + 4.7T - 0.04T^2$
- f is not known exactly
- possible error on the measurement of TDOA (neglected)

INRIA
SOPHIA ANTIPOLIS

Managing uncertainties

- c varies with temperature T
  - in water, $c = 1404.3 + 4.7T - 0.04T^2$
- f is not known exactly
- possible error on the measurement of TDOA (neglected)

effect on the localization ?

Computing the region

Computing the region

- detection if $A_{ij} > \epsilon$

Computing the region

- detection if $A_{ij} > \epsilon$
- $\mathcal{B}_1$: 2D-box including all detected points

Computing the region

- detection if $A_{ij} > \epsilon$
- $\mathcal{B}_1$: 2D-box including all detected points
- $\mathcal{L}$: list of n boxes, initially $\mathcal{L} = \{\mathcal{B}_1\}$

Computing the region

- detection if $A_{ij} > \epsilon$
- $\mathcal{B}_1$: 2D-box including all detected points
- $\mathcal{L}$: list of n boxes, initially $\mathcal{L} = \{\mathcal{B}_1\}$
- $\mathcal{S}$: 4D-box, $\{\mathcal{B}_k, f, c\}$

Computing the region

- detection if $A_{ij} > \epsilon$
- $\mathcal{B}_1$: 2D-box including all detected points
- $\mathcal{L}$: list of n boxes, initially $\mathcal{L} = \{\mathcal{B}_1\}$
- $\mathcal{S}$: 4D-box, $\{\mathcal{B}_k, f, c\}$
- $\mathcal{M}$: list of m 4D-boxes, initially $\mathcal{M} = \{\mathcal{S}_1\}$

INRIA SOPHIA ANTIPOLIS

Computing the region

Algorithm $\texttt{Find}(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

end for

Computing the region

Algorithm $\texttt{Find}(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

end for

Computing the region

Algorithm `Find`$(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

end for

Computing the region

Algorithm `Find`$(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

if $\forall\ i, j\ \underline{F} > \epsilon$ **then**

store $\mathcal{B}_k$ as solution, **next**

end if

end for

INRIA SOPHIA ANTIPOLIS

Computing the region

Algorithm $\texttt{Find}(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

if $\forall\ i, j\ \underline{F} > \epsilon$ **then**

store $\mathcal{B}_k$ as solution, **next**

end if

if Diam($\mathcal{B}_k$)$< \mu$ **then**

neglect $\mathcal{B}_k$,**next**

end if

end for

Computing the region

Algorithm $\texttt{Find}(\mathcal{B}, \mathcal{L}, n)$

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

if $\forall\ i, j\ \underline{F} > \epsilon$ **then**

store $\mathcal{B}_k$ as solution, **next**

end if

if Diam($\mathcal{B}_k$)$< \mu$ **then**

neglect $\mathcal{B}_k$,**next**

end if

bisect $\mathcal{B}_k$, store the result in $\mathcal{L}$, $n = n + 2$

end for

Computing the region

Algorithm `Find`($\mathcal{B}, \mathcal{L}, n$)

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

if $\forall\ i, j\ \underline{F} > \epsilon$ **then**

store $\mathcal{B}_k$ as solution, **next**

end if

if Diam($\mathcal{B}_k$)$< \mu$ **then**

neglect $\mathcal{B}_k$,**next**

end if

bisect $\mathcal{B}_k$, store the result in $\mathcal{L}$, $n = n + 2$

end for

`Loop`($\mathcal{S}, \mathcal{M}, l$): same algorithm than `Find` but

- maximum of l bisection
- returns 1 if all $\mathcal{S}$ boxes have been processed, 0 otherwise

INRIA SOPHIA ANTIPOLIS

Computing the region

Algorithm `Find`($\mathcal{B}, \mathcal{L}, n$)

for $k = 1$ to n **do**

compute all $F_{ij} = A_{ij}(\mathcal{B}_k)$

if $\exists\ i,\ j$ such that $\overline{F_{ij}} < \epsilon$ **then**

next

end if

if $\forall\ i, j\ \underline{F} > \epsilon$ **then**

store $\mathcal{B}_k$ as solution

next

end if

if `Loop`($\mathcal{S}, \mathcal{M}, 100$)=1 **then**

store $\mathcal{B}_k$ as solution, **next**

end if

if Diam($\mathcal{B}_k$)< μ **then**

neglect $\mathcal{B}_k$,**next**

end if

bisect $\mathcal{B}_k$, store the result in $\mathcal{L}$, $n = n + 2$

end for

INRIA SOPHIA ANTIPOLIS

Result

Result

- c in [1465,1496] m/s ($\pm$ 5 degrees variation)

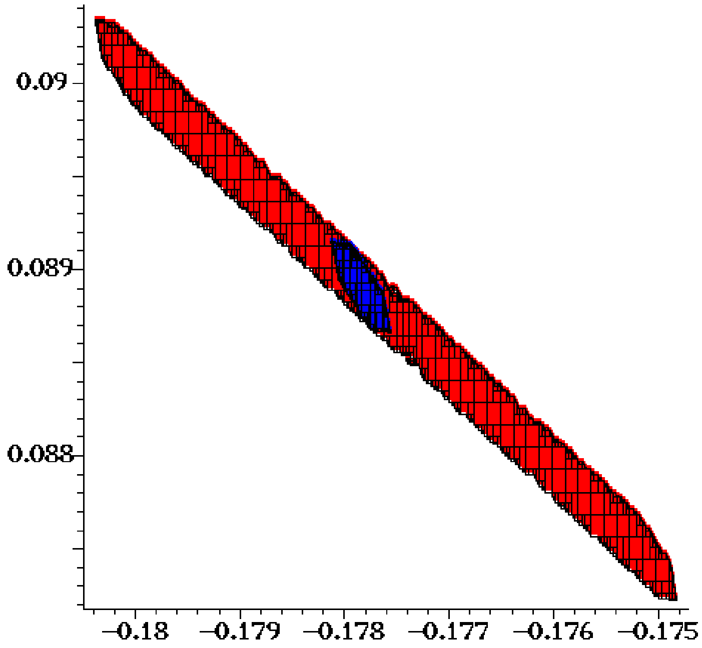
INRIA SOPHIA ANTIPOLIS

Result

- c in [1465,1496] m/s ($\pm$ 5 degrees variation)
- f in [295,305] kHz

Result

- c in [1465,1496] m/s ($\pm$ 5 degrees variation)
- f in [295,305] kHz



No uncertainty, with uncertainties

INRIA SOPHIA ANTIPOLIS

Design problem

Design problem

- two receivers in an exactly known location

Design problem

- two receivers in an exactly known location
- a given workspace $\mathcal{W}$ for the robot

Design problem

- two receivers in an exactly known location
- a given workspace $\mathcal{W}$ for the robot
- given uncertainties on f, c

Design problem

- two receivers in an exactly known location
- a given workspace $\mathcal{W}$ for the robot
- given uncertainties on f, c
- a maximal localization error α

INRIA SOPHIA ANTIPOLIS

Design problem

- two receivers in an exactly known location
- a given workspace $\mathcal{W}$ for the robot
- given uncertainties on f, c
- a maximal localization error α

Find possible location of R_3 so that:

- for all robot location in $\mathcal{W}$ localization error is $< \alpha$

Design problem

- two receivers in an exactly known location
- a given workspace $\mathcal{W}$ for the robot
- given uncertainties on f, c
- a maximal localization error α

Find possible location of R_3 so that:

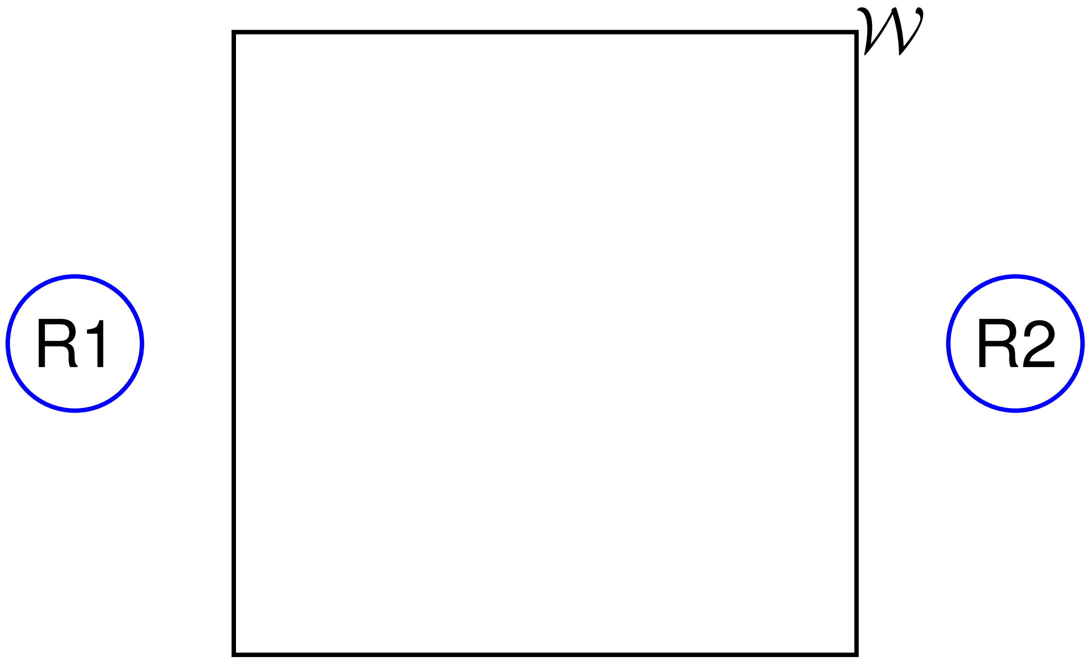
- for all robot location in $\mathcal{W}$ localization error is $< \alpha$
- allowing to manage uncertainties on the location of R_3

INRIA SOPHIA ANTIPOLIS

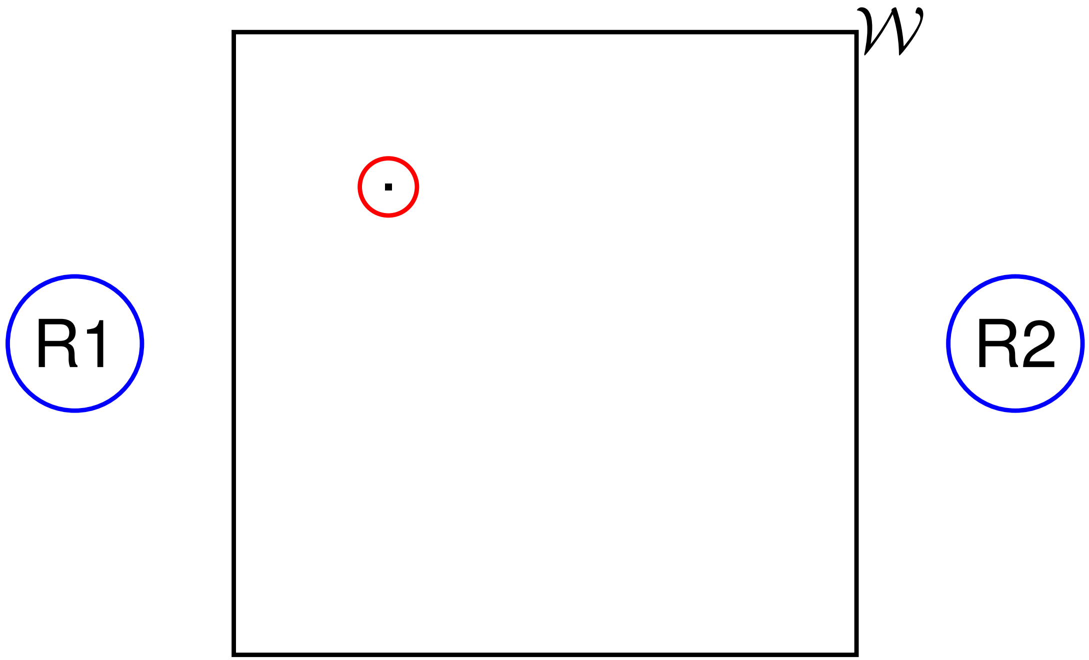
Approach

Approach

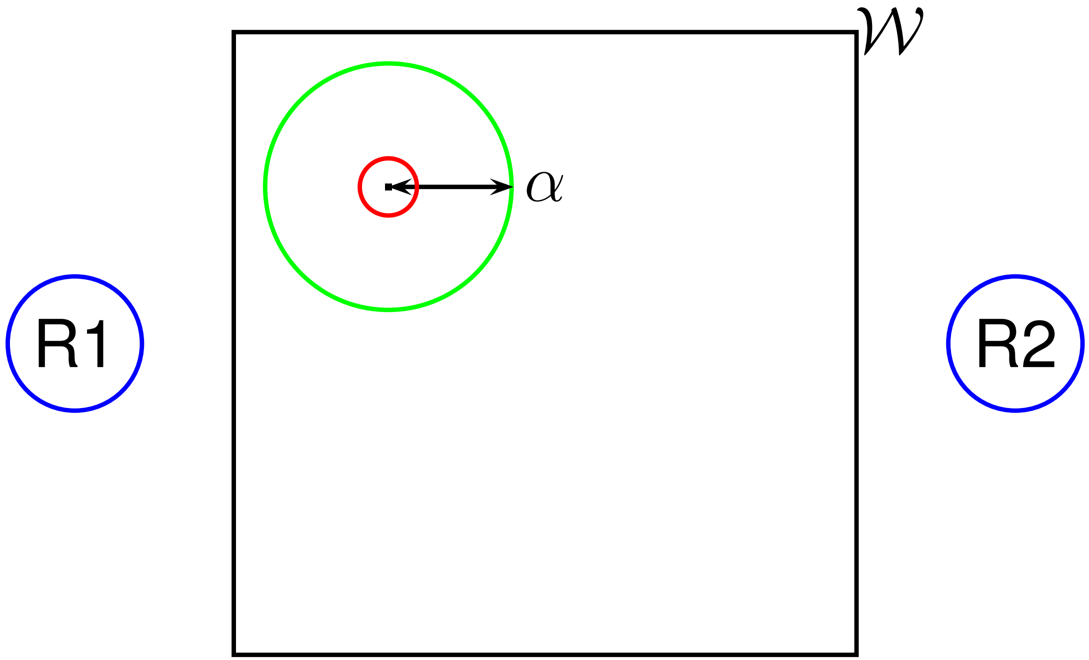
Approach



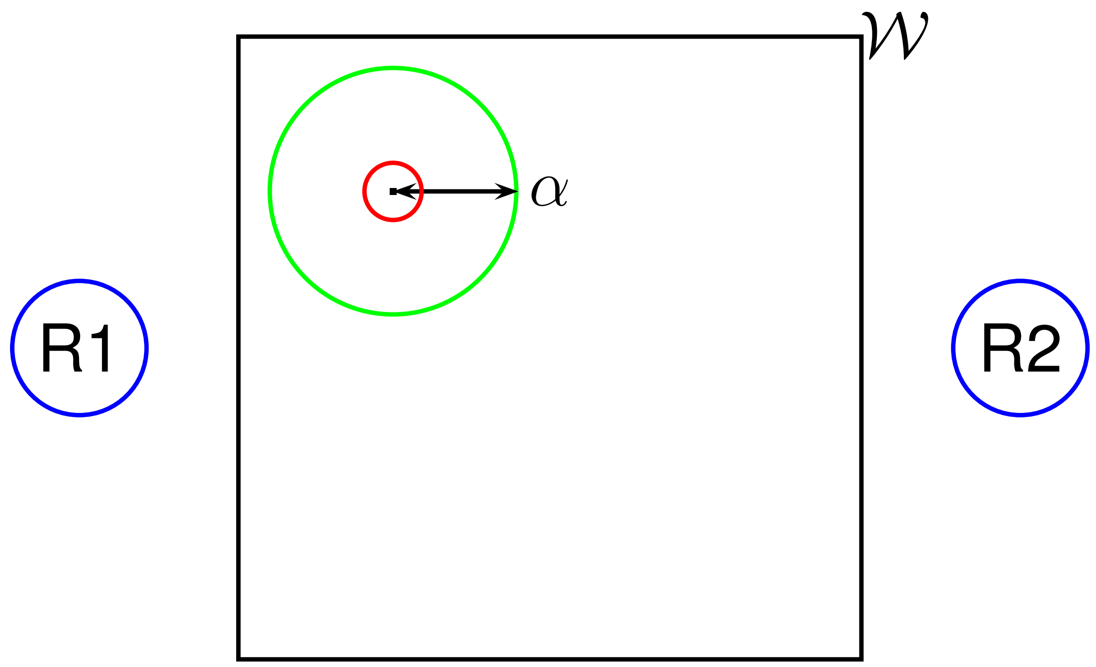
Approach



Approach



Approach



If circle centered at P, radius α has no detectable points

$\Rightarrow$ localization error at P is $< \alpha$

INRIA SOPHIA ANTIPOLIS

Algorithm

Algorithm

- Upper loop: box with x, y (coordinates of R_3)

INRIA SOPHIA ANTIPOLIS

Algorithm

- Upper loop: box with x, y (coordinates of R_3)
- Inner loop: box with

Algorithm

- Upper loop: box with x, y (coordinates of R_3)
- Inner loop: box with
  - f, c

Algorithm

- Upper loop: box with x, y (coordinates of R_3)
- Inner loop: box with
  - f, c
  - x_P, y_P: coordinates of P in $\mathcal{W}$

Algorithm

- Upper loop: box with x, y (coordinates of R_3)
- Inner loop: box with
  - f, c
  - x_P, y_P: coordinates of P in $\mathcal{W}$
  - θ: angle on the circle centered at P, radius α

INRIA SOPHIA ANTIPOLIS

Algorithm

- Upper loop: box with x, y (coordinates of R_3)
- Inner loop: box with
  - f, c
  - x_P, y_P: coordinates of P in $\mathcal{W}$
  - θ: angle on the circle centered at P, radius α

computationally expensive

INRIA SOPHIA ANTIPOLIS

Algorithms

Algorithms

Algorithm 1: check only specific points of $\mathcal{W}$

Algorithms

Algorithm 1: check only specific points of $\mathcal{W}$

$$\Downarrow$$

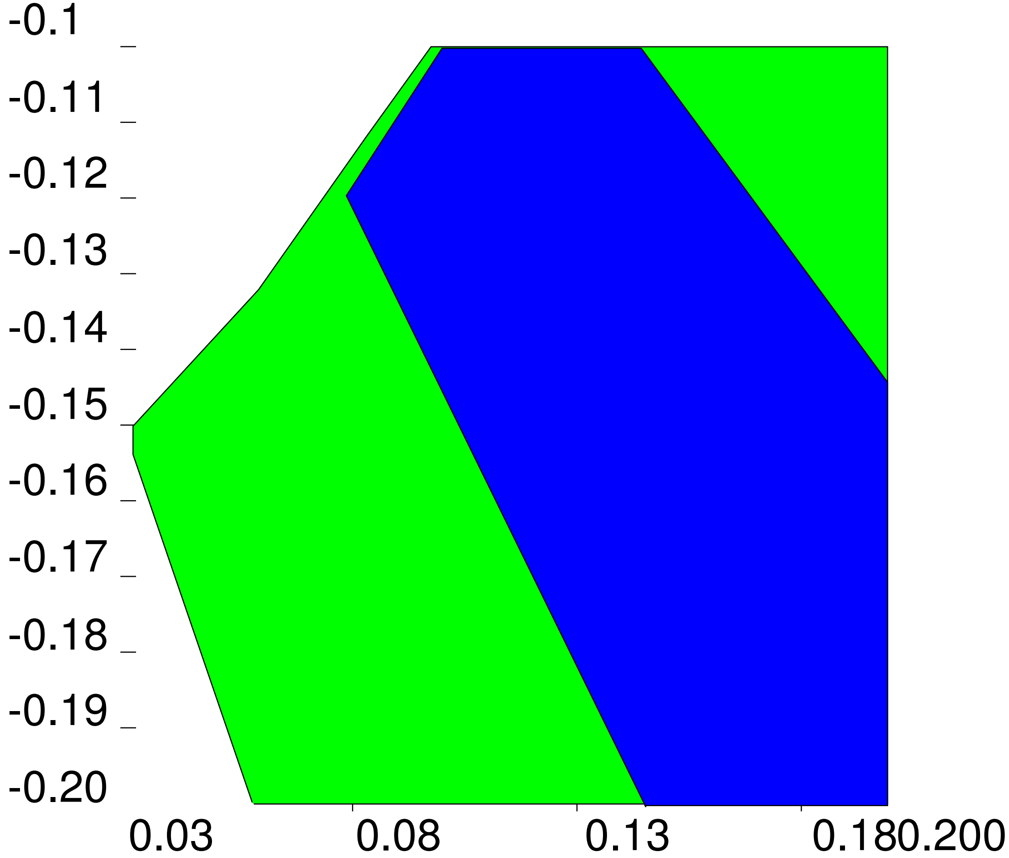
result is an over-approximation of the possible region for R_3

Algorithms

Algorithm 1: check only specific points of $\mathcal{W}$

$$\Downarrow$$

result is an over-approximation of the possible region for R_3



Algorithms

Algorithm 1: check only specific points of $\mathcal{W}$

$$\Downarrow$$

result is an over-approximation of the possible region for R_3

computationaly expensive: 20 hours on 17 computers

INRIA SOPHIA ANTIPOLIS

Algorithms

Algorithms

Algorithm 2:

- select a possible location for R_3 within the result of Algorithm 1
- choose a positioning accuracy for R_3
- check the whole workspace for accuracy

Algorithms

Algorithm 2:

- select a possible location for R_3 within the result of Algorithm 1
- choose a positioning accuracy for R_3
- check the whole workspace for accuracy

location 0.123$\pm$ 0.0005, -0.123 $\pm$ 0.005 is valid

Conclusion

Conclusion

Analysis

- more realistic localization (inner and outer approximation)

Conclusion

Analysis

- more realistic localization (inner and outer approximation)

Synthesis

- design for given performances of the system

Conclusion

Analysis

- more realistic localization (inner and outer approximation)

Synthesis

- design for given performances of the system

Prospective

Conclusion

Analysis

- more realistic localization (inner and outer approximation)

Synthesis

- design for given performances of the system

Prospective

- better signal model (reflection, . . .)

INRIA
SOPHIA ANTIPOLIS

Conclusion

Analysis

- more realistic localization (inner and outer approximation)

Synthesis

- design for given performances of the system

Prospective

- better signal model (reflection, . . .)
- inaccuracies on the location of R_1, R_2

INRIA SOPHIA ANTIPOLIS