An introduction to THE STATE CONDITIONAL FILTERING PARADIGM

Speaker: Algo Carè (University of Brescia, IT)

In collaboration with:
Marco C. Campi (University of Brescia, IT)
Erik Weyer (The University of Melbourne, VIC, AU)

ERNSI workshop 2024
September 30 - October 2

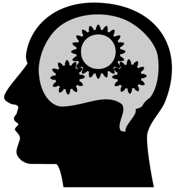
An introduction to THE STATE CONDITIONAL FILTERING PARADIGM

Main reference:

"State Conditional Filtering,"

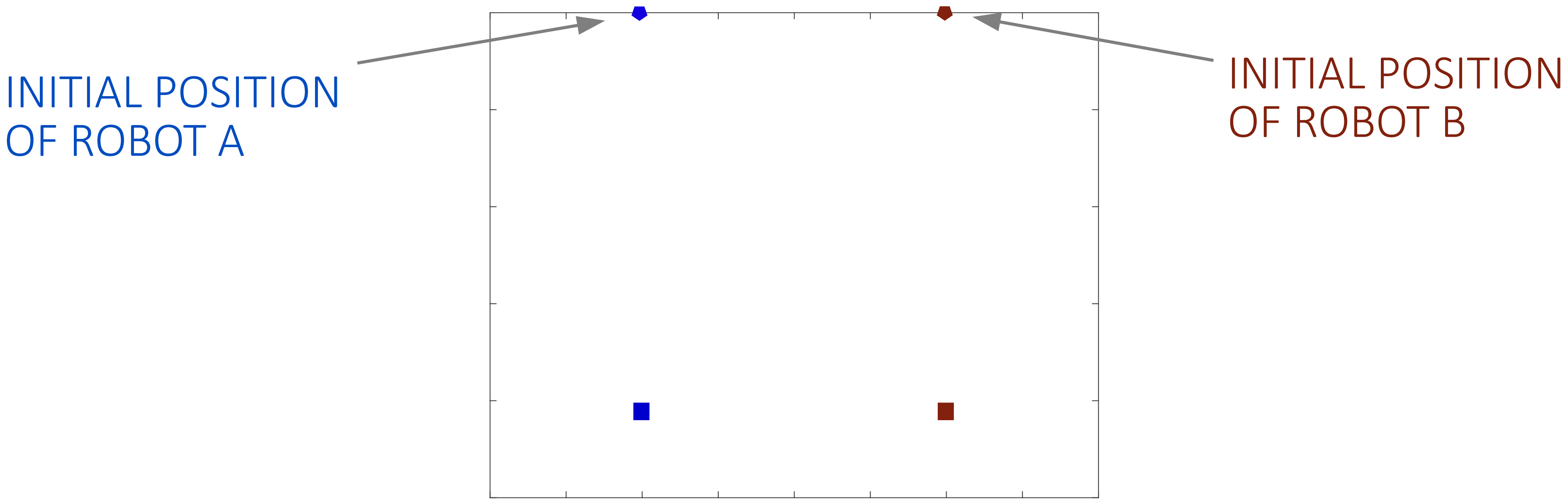
IEEE Transactions on Automatic Control, 67(7):3381-3395, 2022

A. Carè, M.C. Campi, E. Weyer

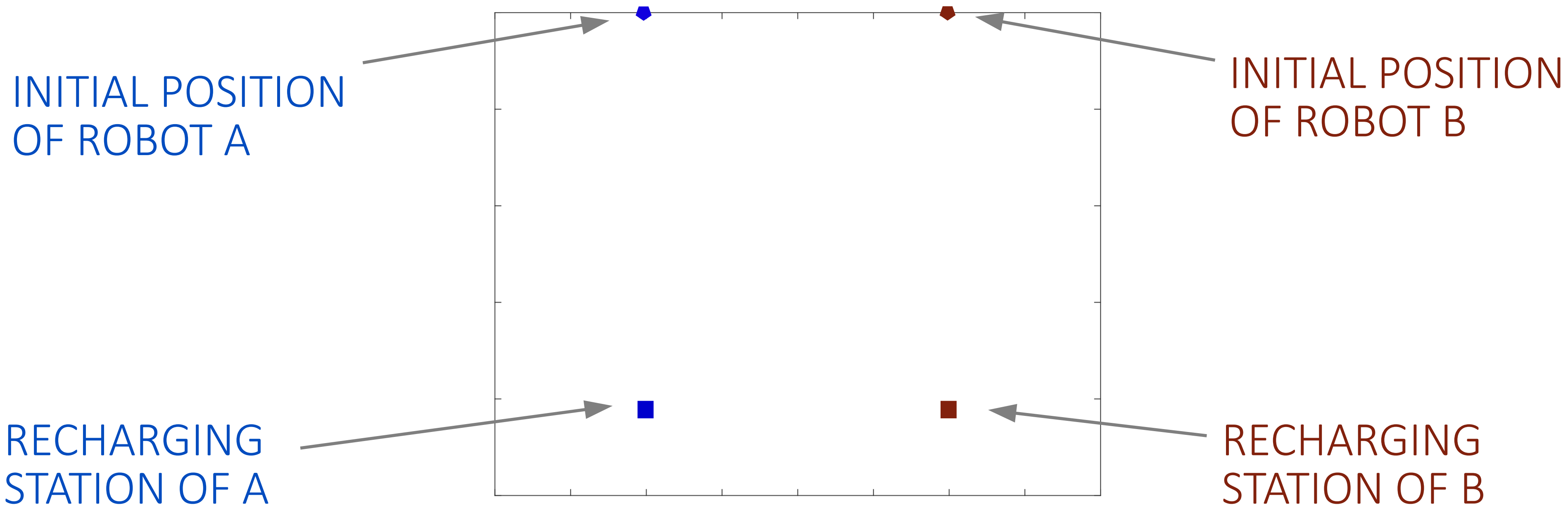


...and some work in progress.

ROBOTICS EXAMPLE
Two robots run their **return-to-base** program

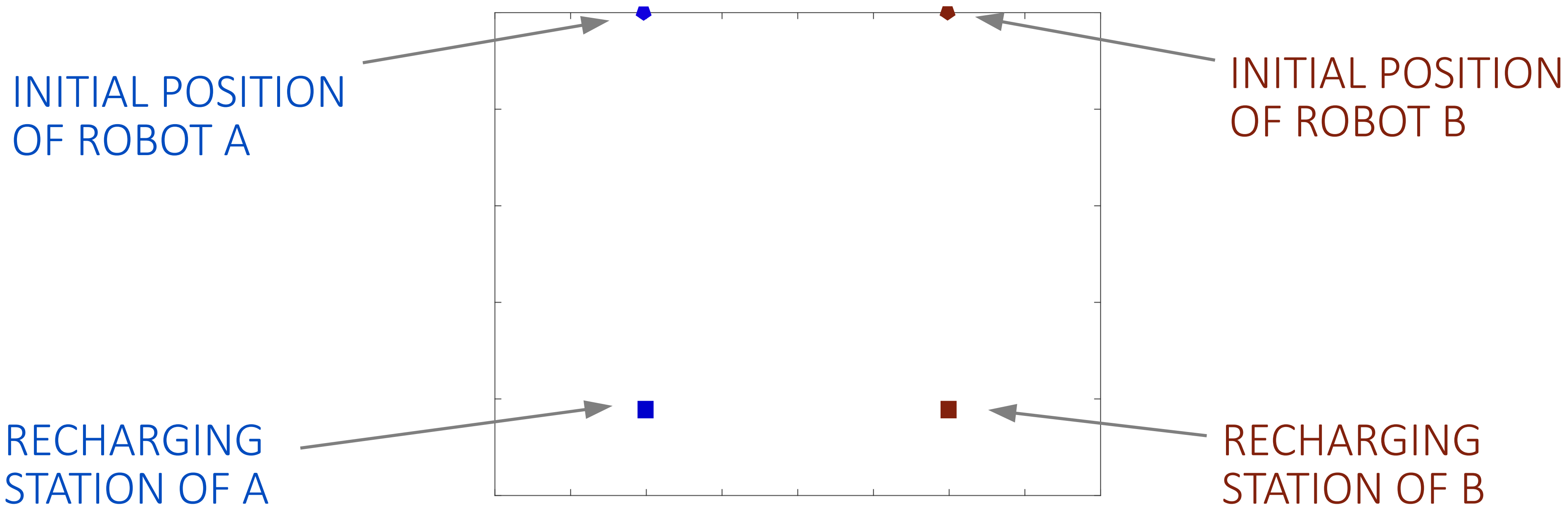


ROBOTICS EXAMPLE
Two robots run their **return-to-base** program



ROBOTICS EXAMPLE
Two robots run their **return-to-base** program

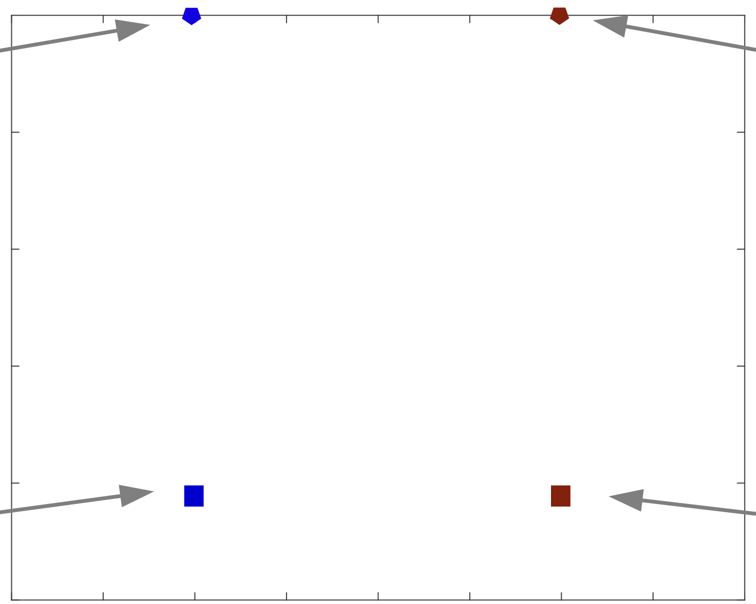
Goal:



ROBOTICS EXAMPLE

Two robots run their **return-to-base** program

Goal: monitor the robots and predict collisions



INITIAL POSITION OF ROBOT A

INITIAL POSITION OF ROBOT B

RECHARGING STATION OF A

RECHARGING STATION OF B

UNCERTAIN DISCRETE-TIME LINEAR SYSTEM

$$\begin{bmatrix} p_{xA,t+1} \\ p_{yA,t+1} \\ u_{xA,t+1} \\ u_{yA,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta & 0 \\ 0 & -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta \end{bmatrix} \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \\ u_{xA,t} \\ u_{yA,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_p \bar{p}_{xA} \\ K_p \bar{p}_{yA} \end{bmatrix} \Delta + v_{A,t} \cdot \Delta,$$

$$y_{A,t} = \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \end{bmatrix} + w_{A,t}$$

UNCERTAIN DISCRETE-TIME LINEAR SYSTEM

$$\begin{bmatrix} p_{xA,t+1} \\ p_{yA,t+1} \\ u_{xA,t+1} \\ u_{yA,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta & 0 \\ 0 & -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta \end{bmatrix} \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \\ u_{xA,t} \\ u_{yA,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_p \bar{p}_{xA} \\ K_p \bar{p}_{yA} \end{bmatrix} \Delta + v_{A,t} \cdot \Delta,$$

stochastic input process

$$\mathrm{V}_A = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0.04^2 & 0 \\ 0 & 0 & 0 & 0.04^2 \end{bmatrix}$$

$$y_{A,t} = \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \end{bmatrix} + w_{A,t}$$

noise process

$$W_A = 0.05^2 I$$

UNCERTAIN DISCRETE-TIME LINEAR SYSTEM

$$\begin{bmatrix} p_{xA,t+1} \\ p_{yA,t+1} \\ u_{xA,t+1} \\ u_{yA,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta & 0 \\ 0 & -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta \end{bmatrix} \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \\ u_{xA,t} \\ u_{yA,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_p \bar{p}_{xA} \\ K_p \bar{p}_{yA} \end{bmatrix} \Delta + v_{A,t} \cdot \Delta,$$

stochastic input process

$$V_A = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0.04^2 & 0 \\ 0 & 0 & 0 & 0.04^2 \end{bmatrix}$$

$$y_{A,t} = \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \end{bmatrix} + w_{A,t}$$

noise process

$$W_A = 0.05^2 I$$

Assumptions: white processes,
zero-mean,
known covariance matrices,
uncorrelated with each other and the initial state,

UNCERTAIN DISCRETE-TIME LINEAR SYSTEM

$$\begin{bmatrix} p_{xA,t+1} \\ p_{yA,t+1} \\ u_{xA,t+1} \\ u_{yA,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta & 0 \\ 0 & -K_p \cdot \Delta & 0 & 1 - K_u \cdot \Delta \end{bmatrix} \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \\ u_{xA,t} \\ u_{yA,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_p \bar{p}_{xA} \\ K_p \bar{p}_{yA} \end{bmatrix} \Delta + v_{A,t} \cdot \Delta,$$

stochastic input process

$$V_A = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0.04^2 & 0 \\ 0 & 0 & 0 & 0.04^2 \end{bmatrix}$$

$$y_{A,t} = \begin{bmatrix} p_{xA,t} \\ p_{yA,t} \end{bmatrix} + w_{A,t}$$

noise process

$$W_A = 0.05^2 I$$

Assumptions:
- white processes,
- zero-mean,
- known covariance matrices,
- uncorrelated with each other and the initial state,
- jointly Gaussian

KALMAN FILTER

KF provides an estimate $\hat{x}_t$ of the system state variables x_t

KALMAN FILTER

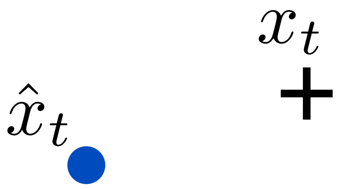
KF provides an estimate $\hat{x}_t$ of the system state variables x_t

x_t

+

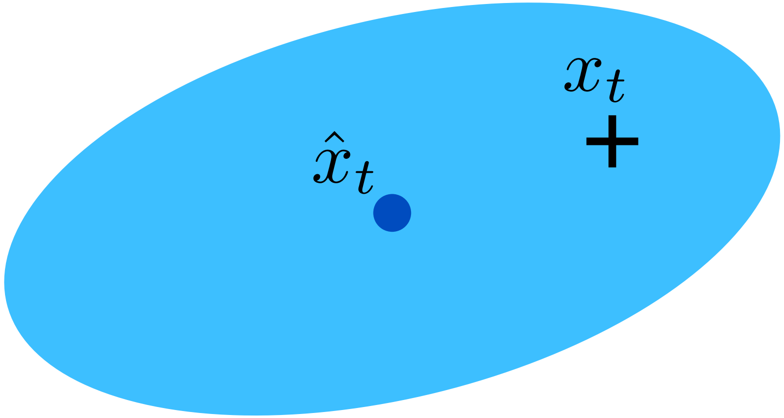
KALMAN FILTER

KF provides an estimate $\hat{x}_t$ of the system state variables x_t



KALMAN FILTER

KF provides an estimate $\hat{x}_t$ of the system state variables x_t

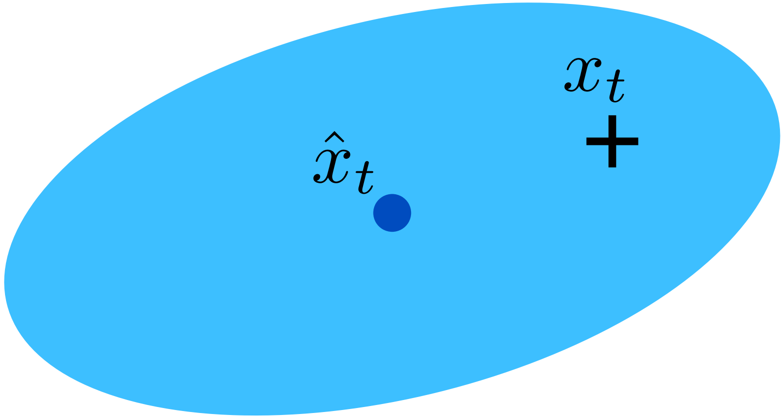


data-independent
size and shape

In the Gaussian setup,
we can construct a 95% probability region

KALMAN FILTER

KF provides an estimate $\hat{x}_t$ of the system state variables x_t

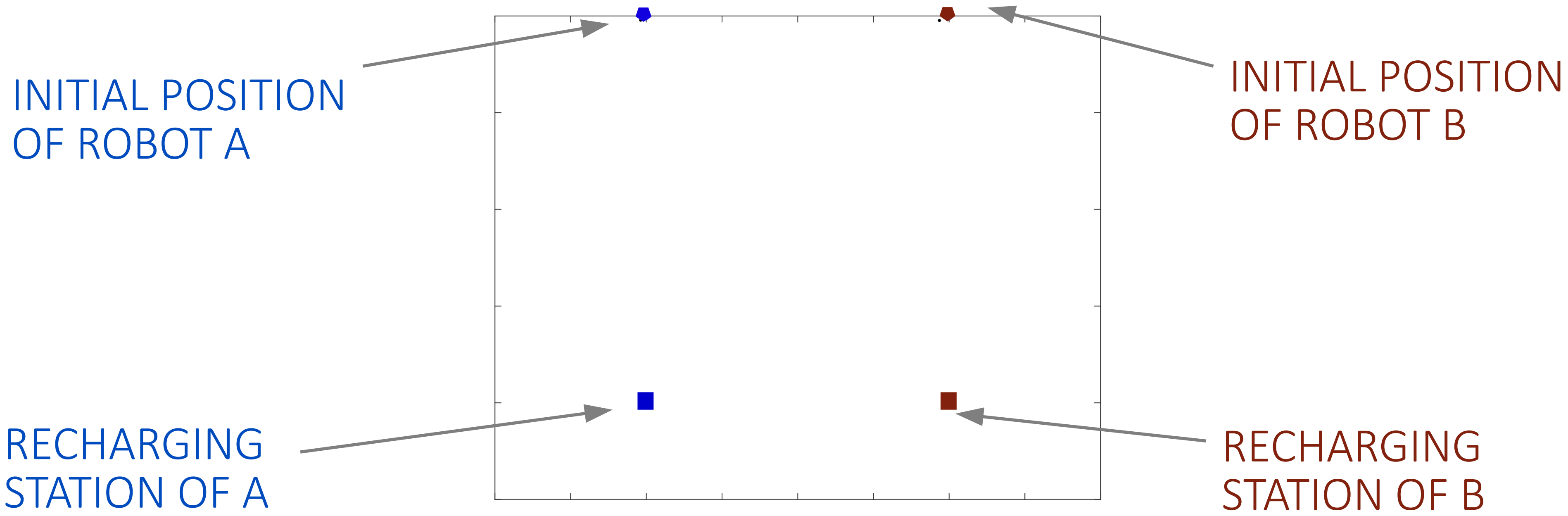


data-independent
size and shape

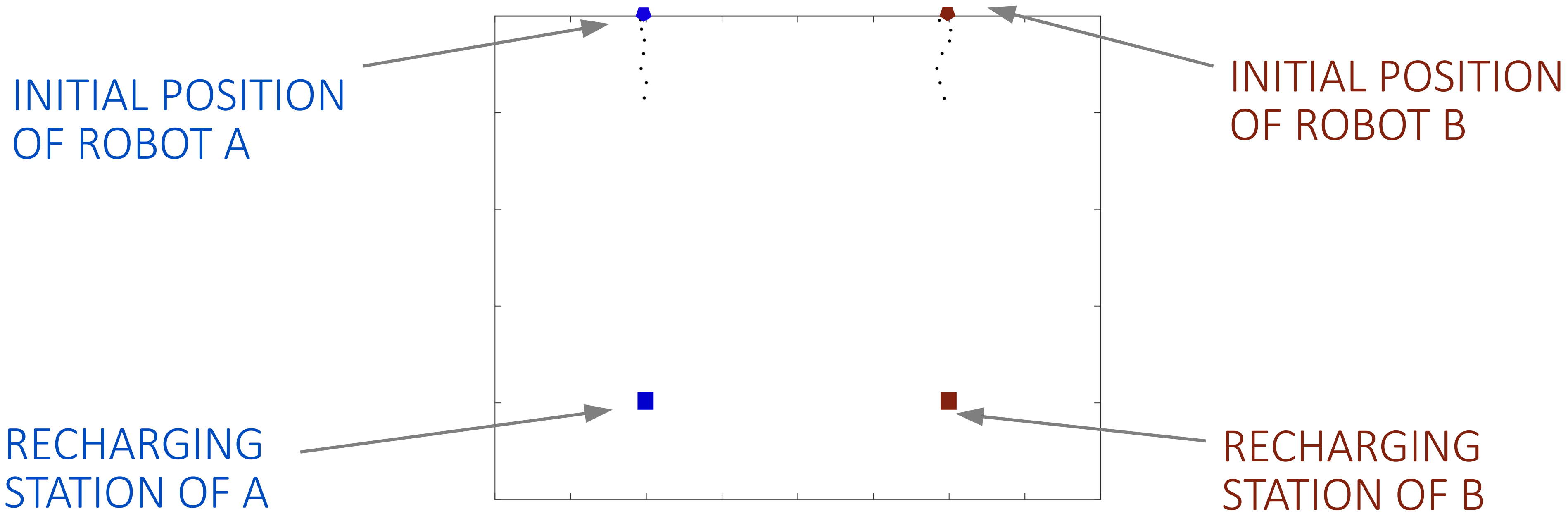
In the Gaussian setup,
we can construct a 95% probability region

Is KF effective for our purpose?

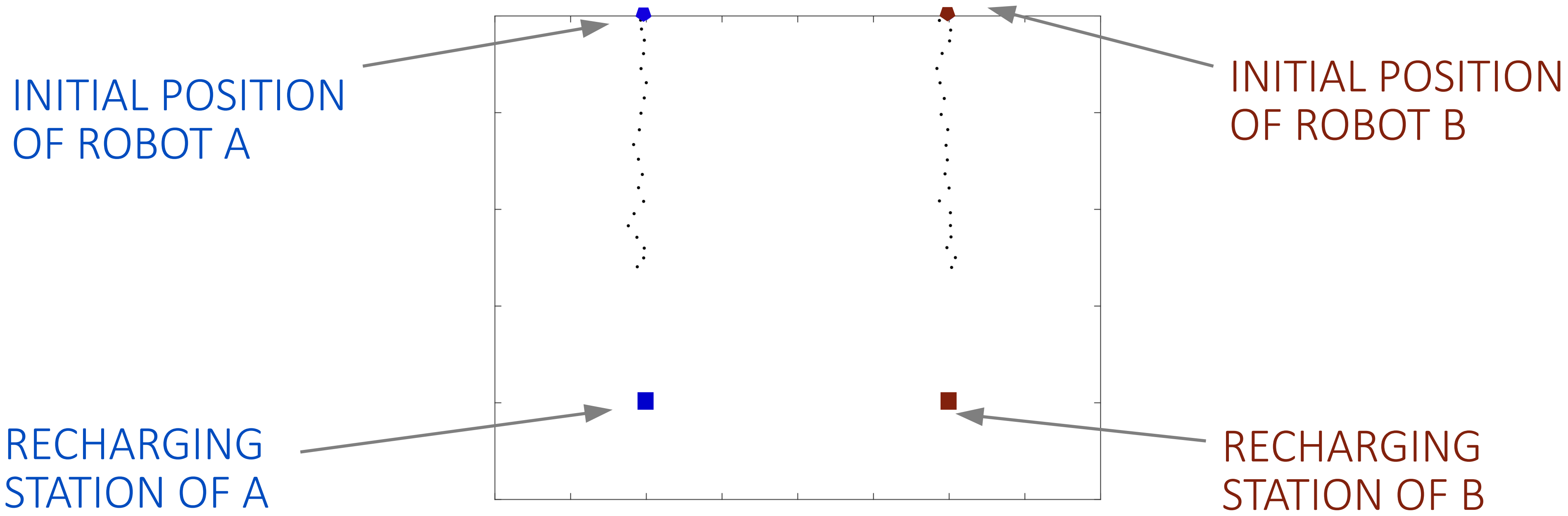
TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program



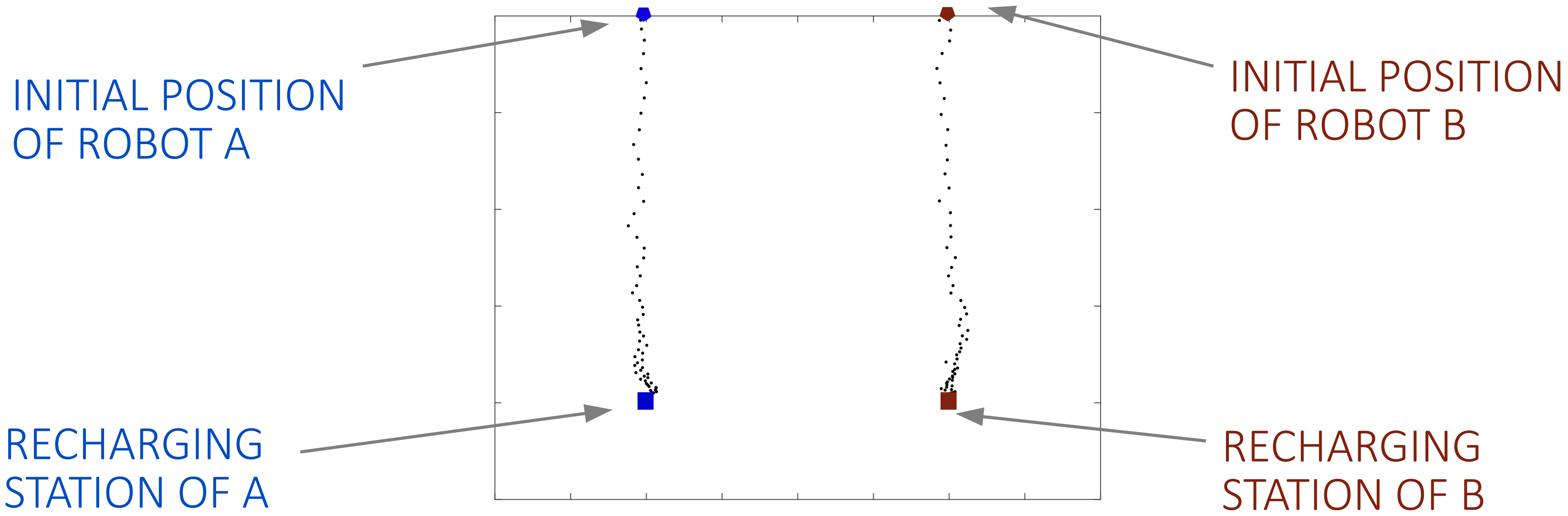
TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program



TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program



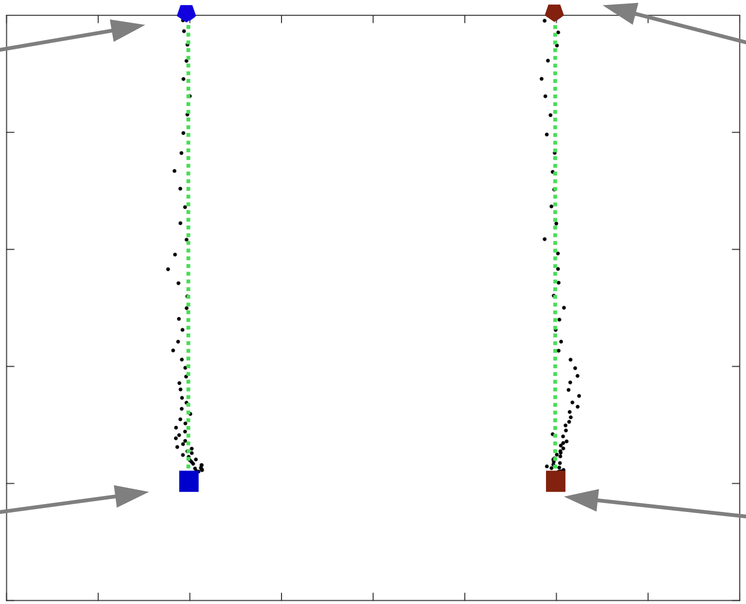
TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program



TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program

INITIAL POSITION
OF ROBOT A

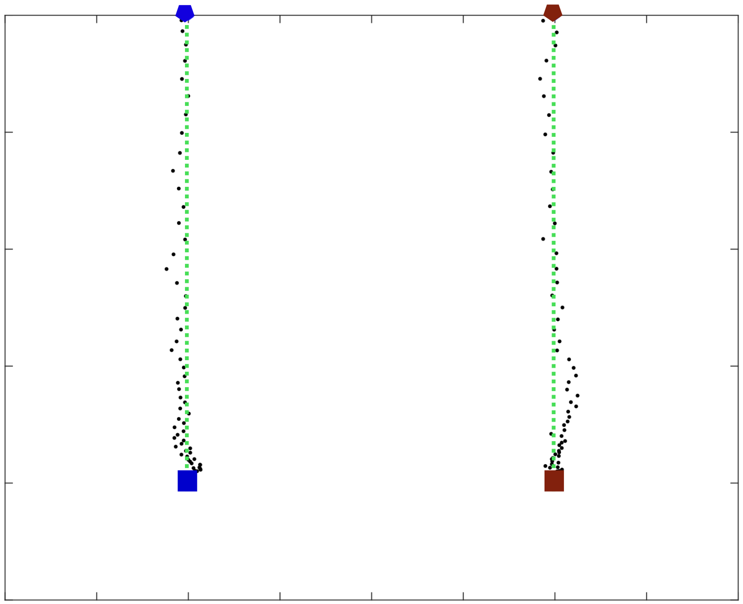
INITIAL POSITION
OF ROBOT B



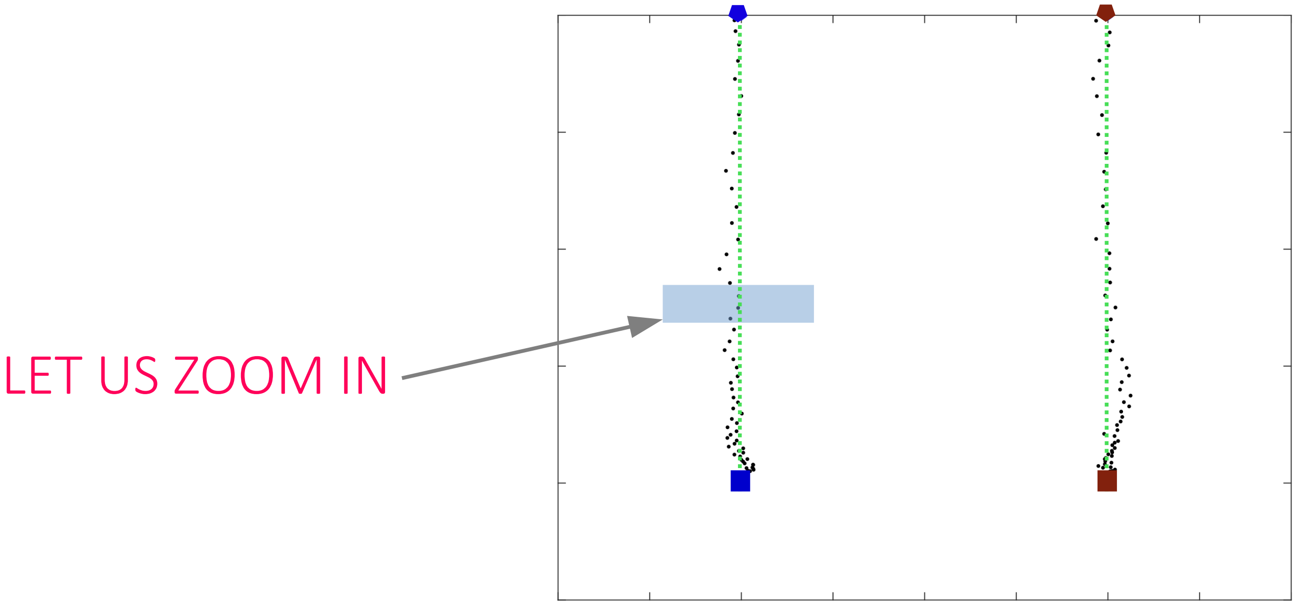
RECHARGING
STATION OF A

RECHARGING
STATION OF B

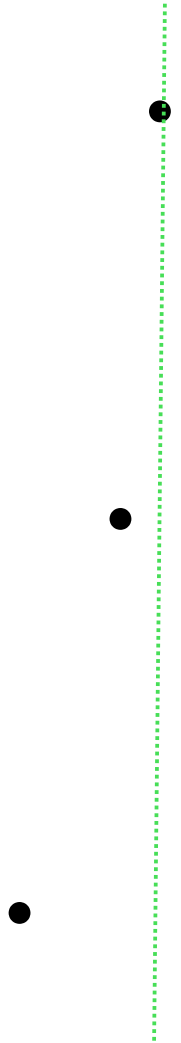
TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program



TYPICAL OBSERVATIONS
Two robots run their **return-to-base** program

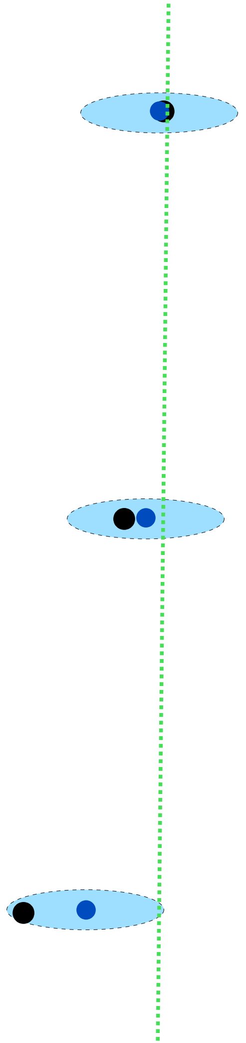


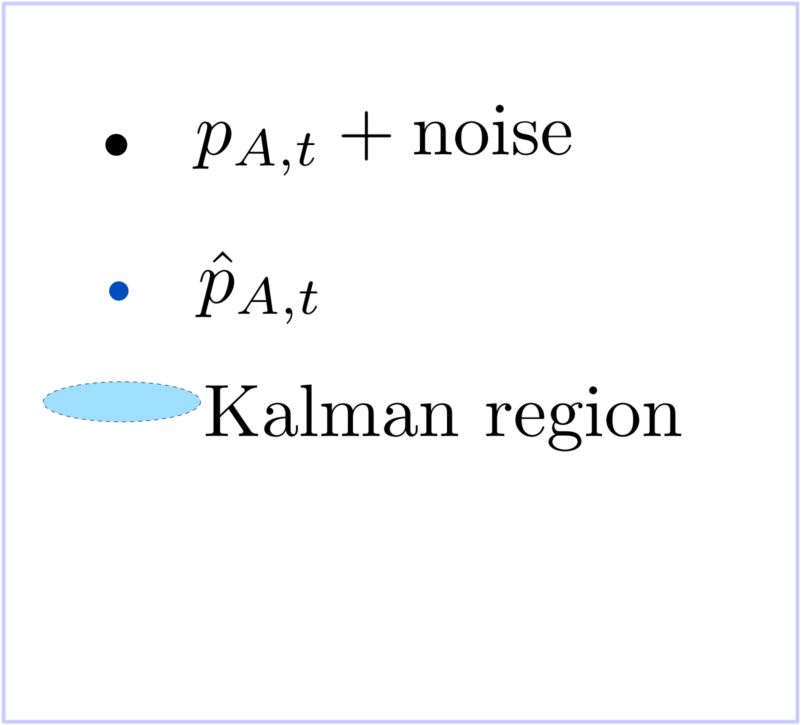
LET US LOOK AT OBSERVATIONS AT $t = 18,\ 19,\ 20$



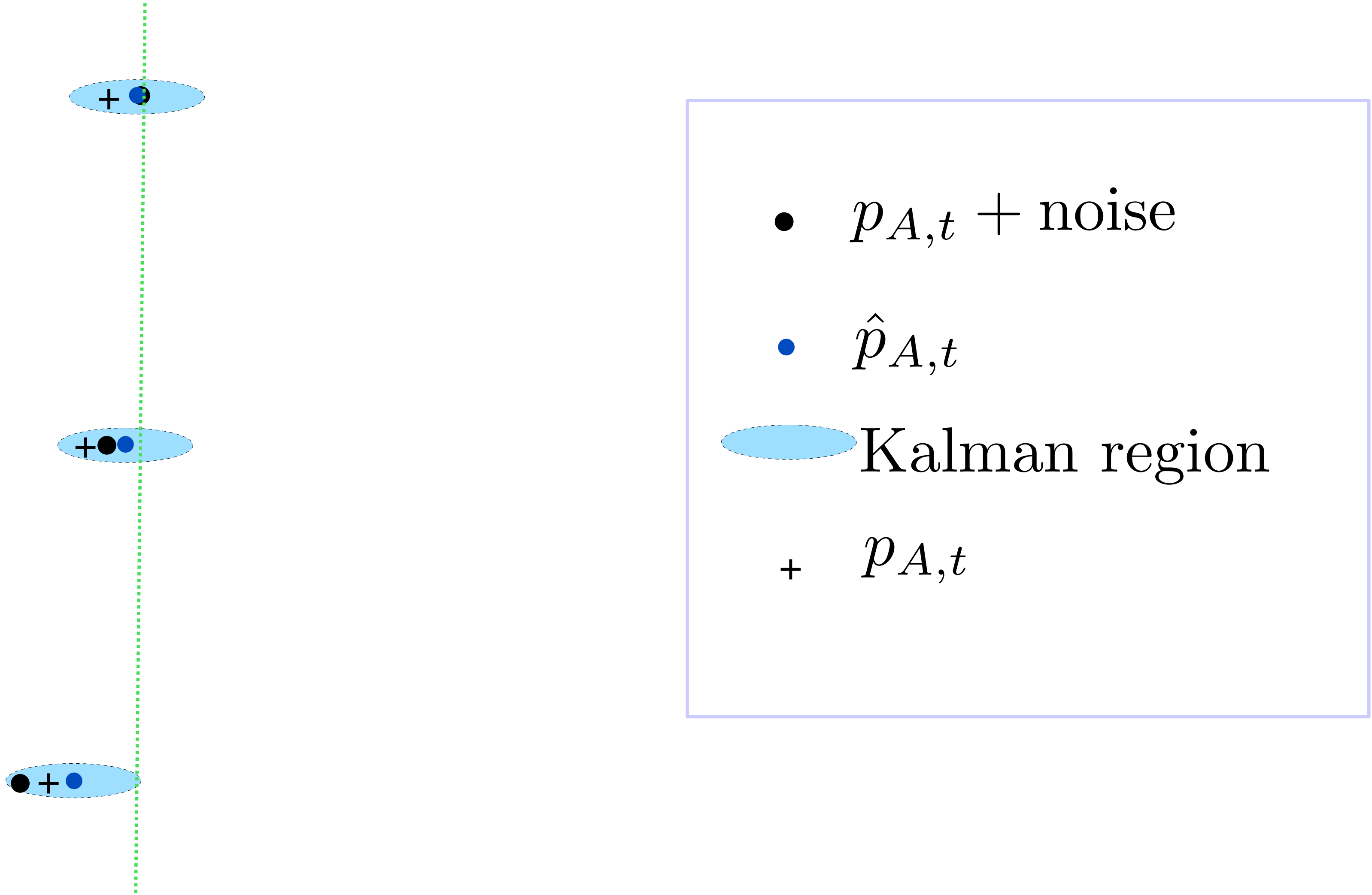
- $p_{A,t} + \text{noise}$

LET US LOOK AT OBSERVATIONS AT $t = 18, 19, 20$

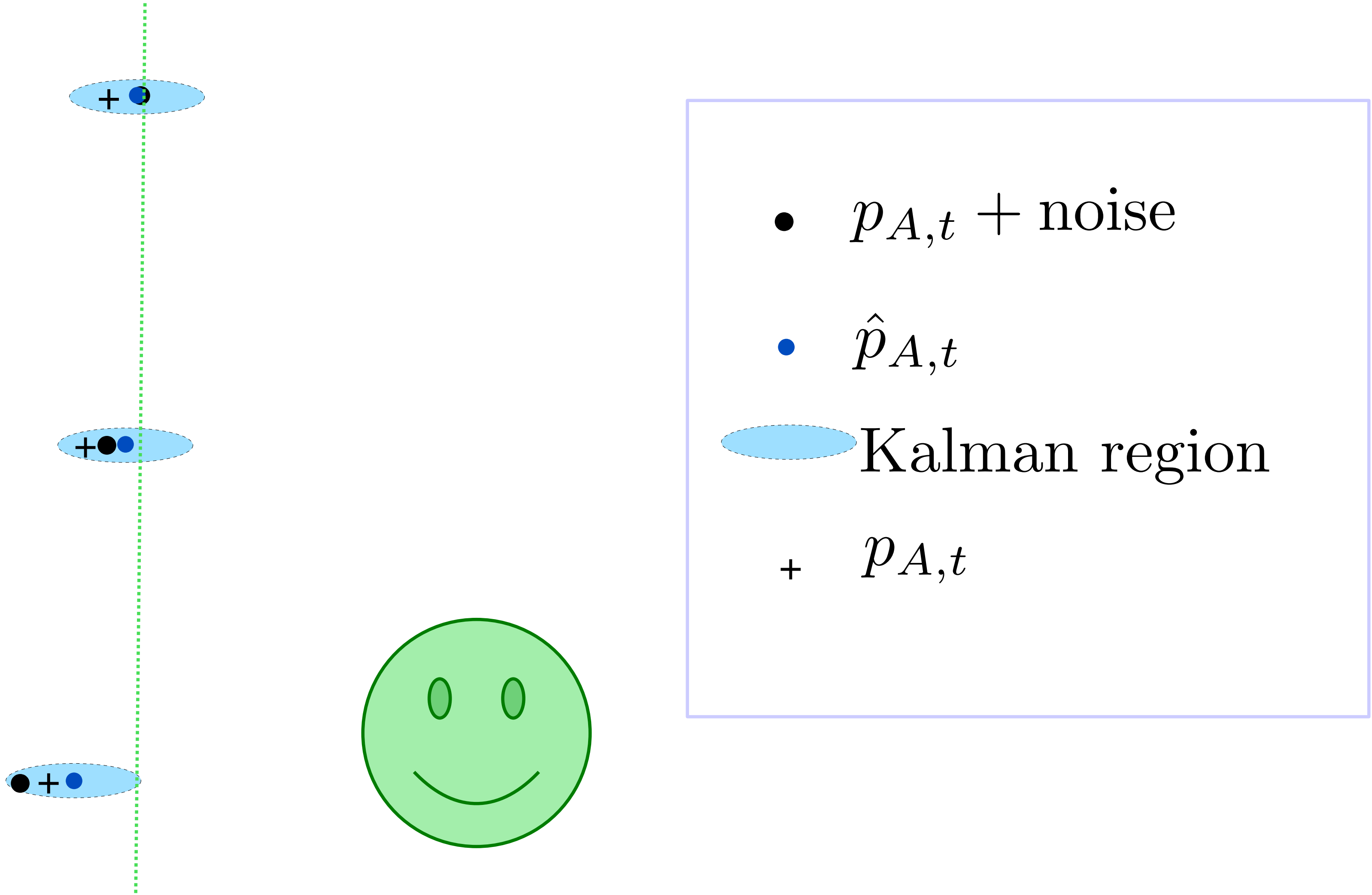




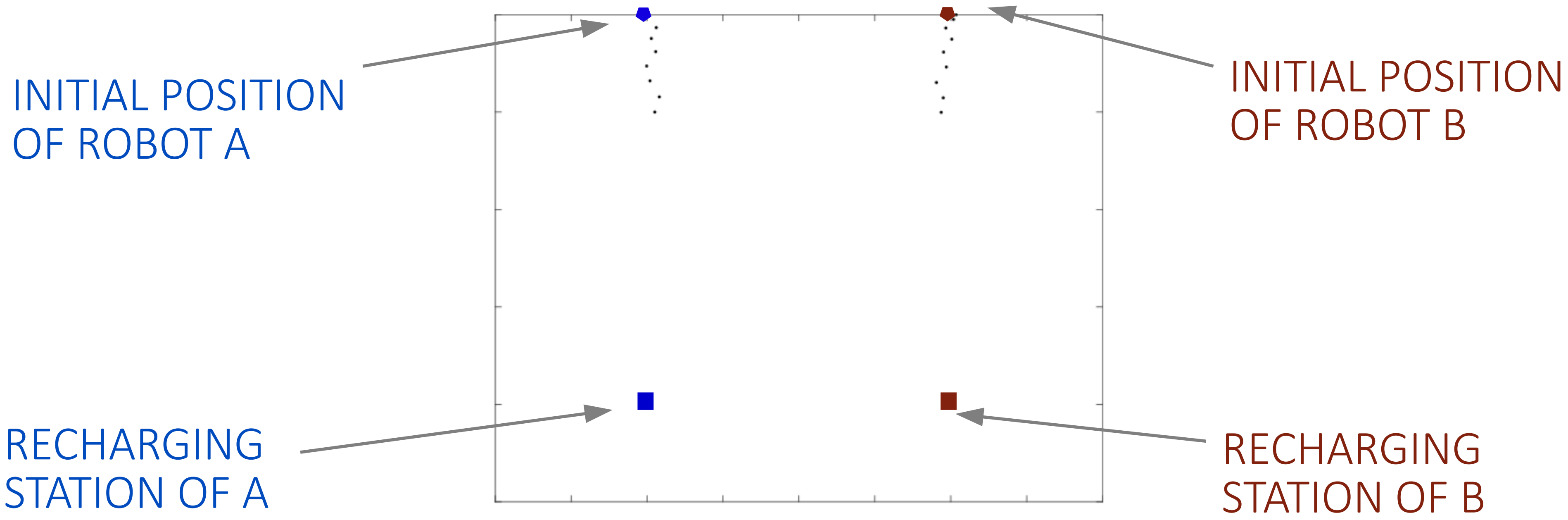
LET US LOOK AT OBSERVATIONS AT $t = 18, 19, 20$



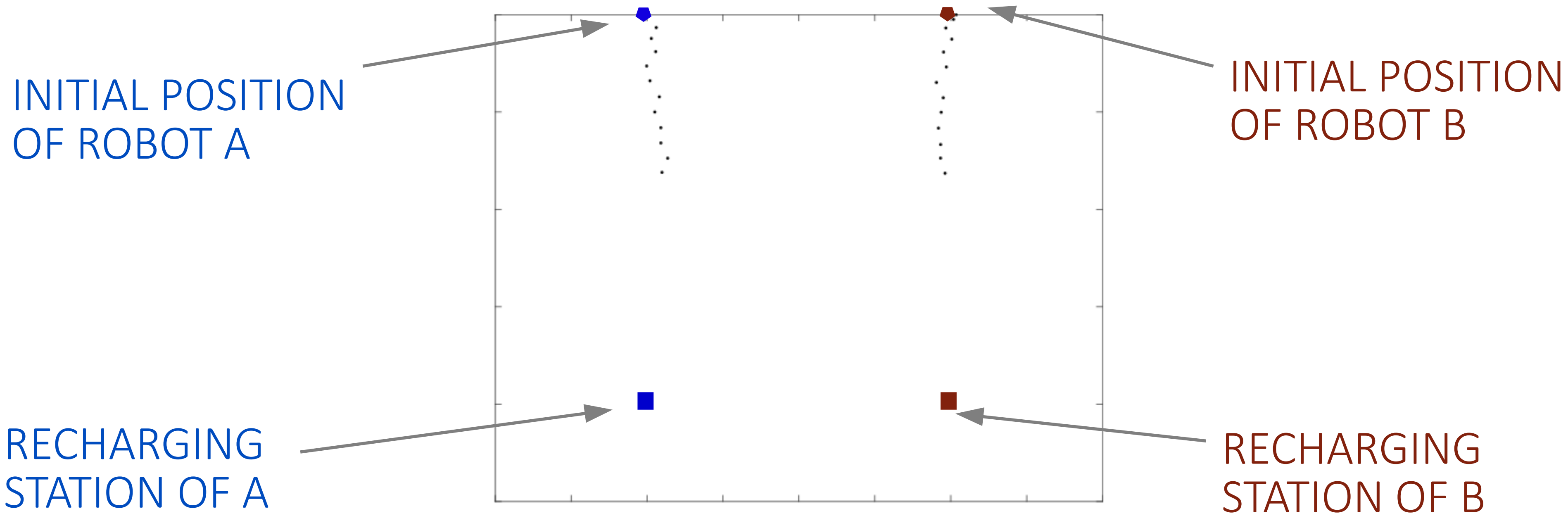
LET US LOOK AT OBSERVATIONS AT $t = 18,\ 19,\ 20$



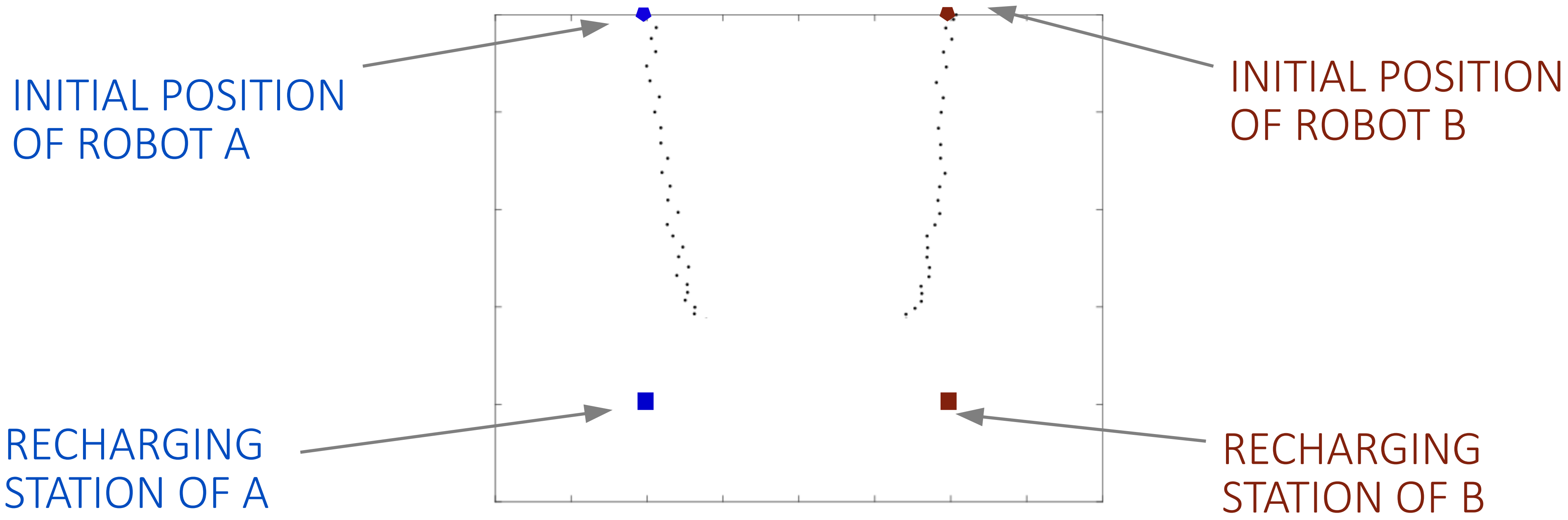
ANOTHER RUN
Two robots run their **return-to-base** program



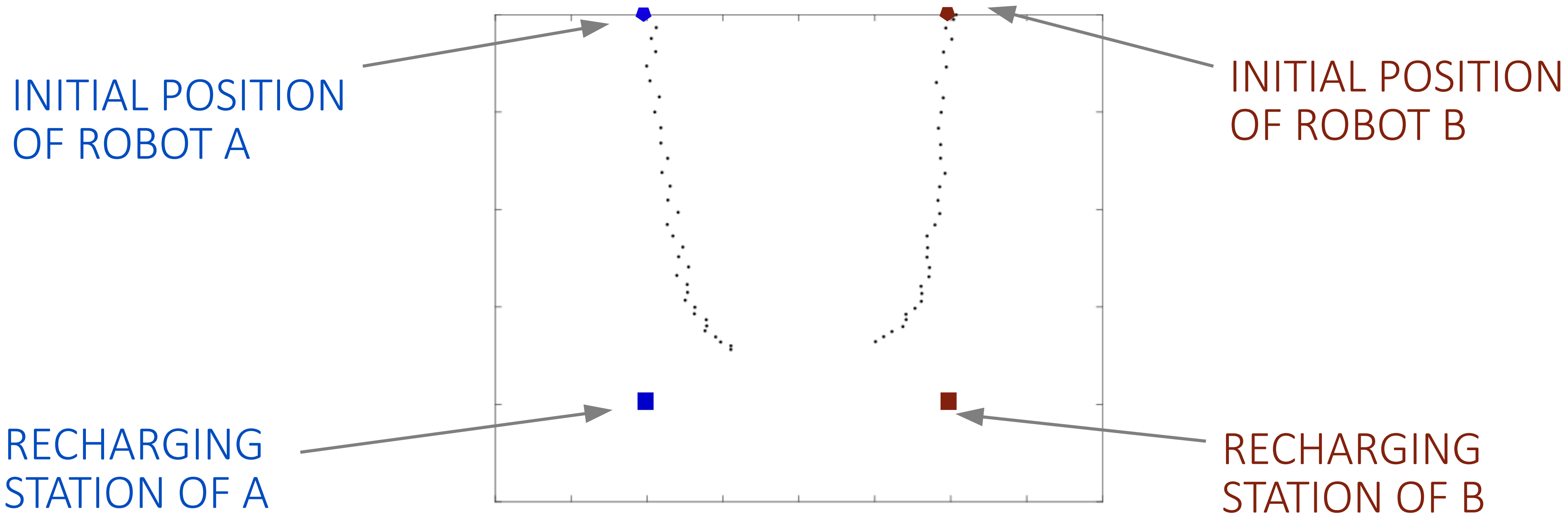
ANOTHER RUN
Two robots run their **return-to-base** program



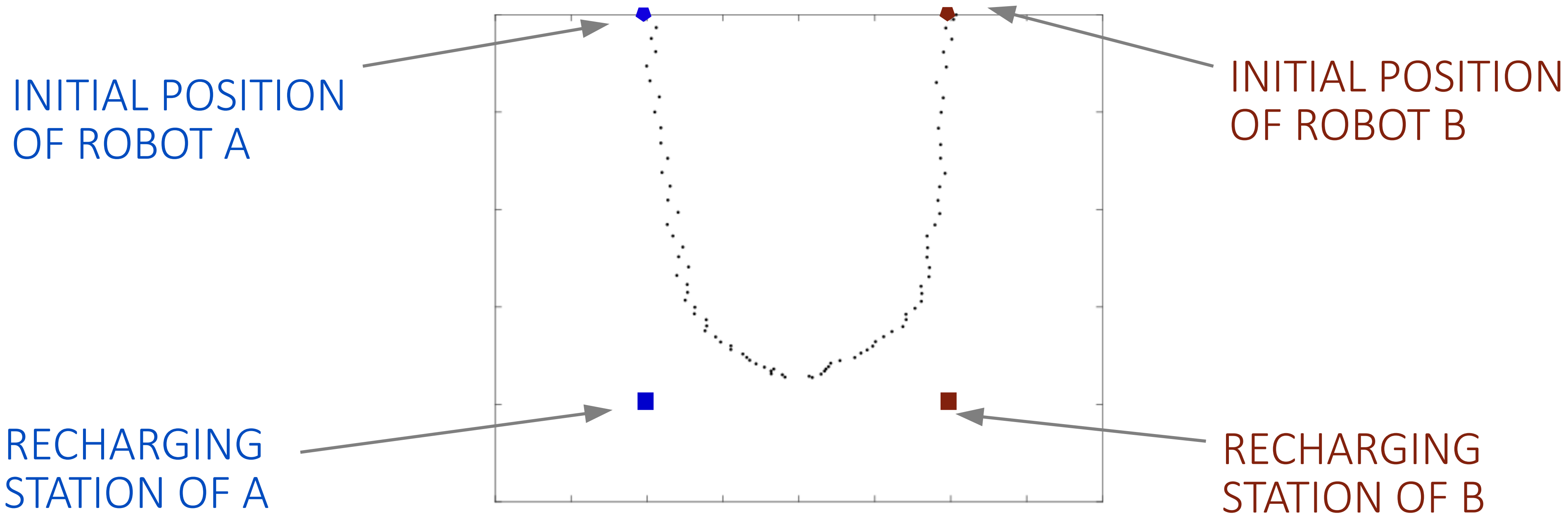
ANOTHER RUN
Two robots run their **return-to-base** program



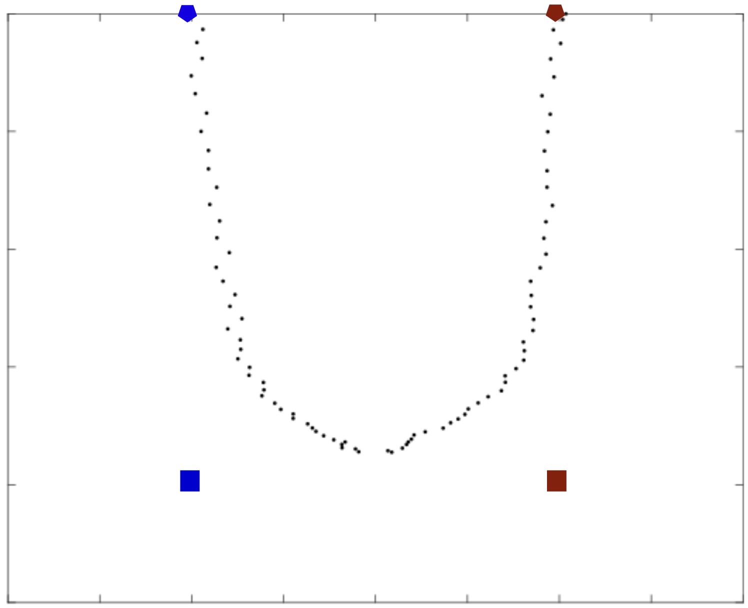
ANOTHER RUN
Two robots run their **return-to-base** program



ANOTHER RUN
Two robots run their **return-to-base** program

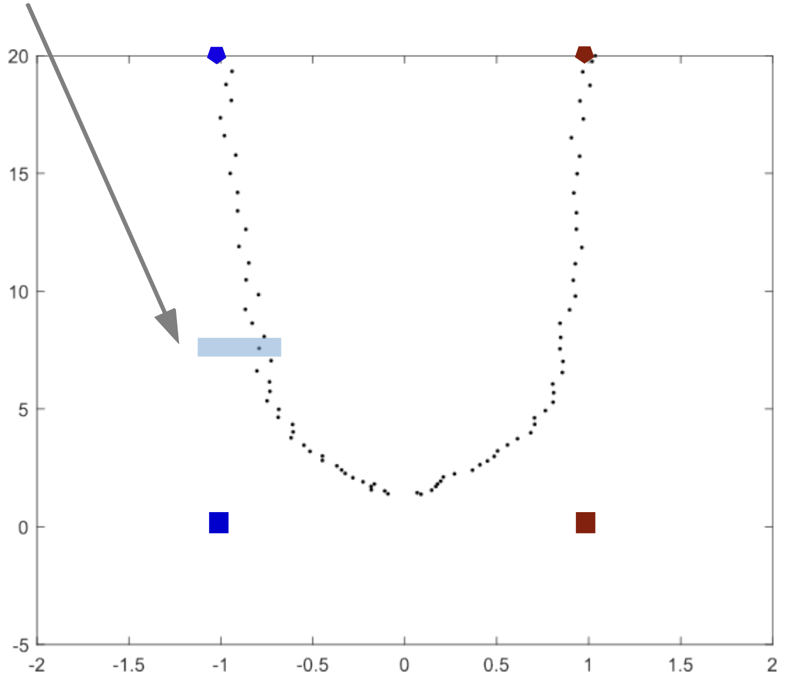


ANOTHER RUN
Two robots run their **return-to-base** program



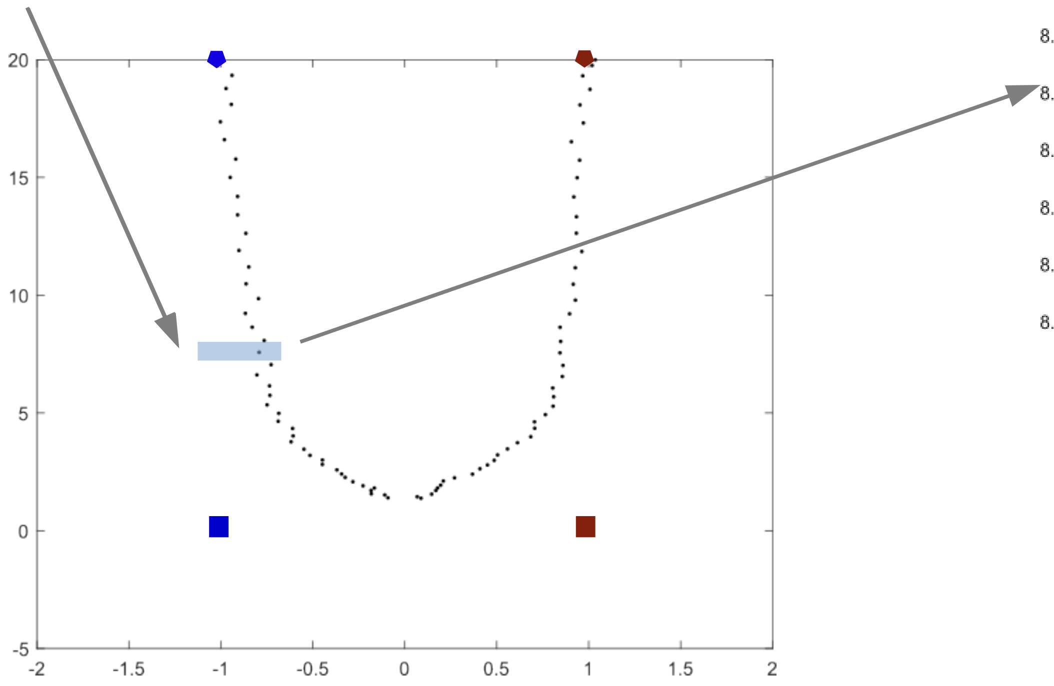
ANOTHER RUN
Two robots run their **return-to-base** program

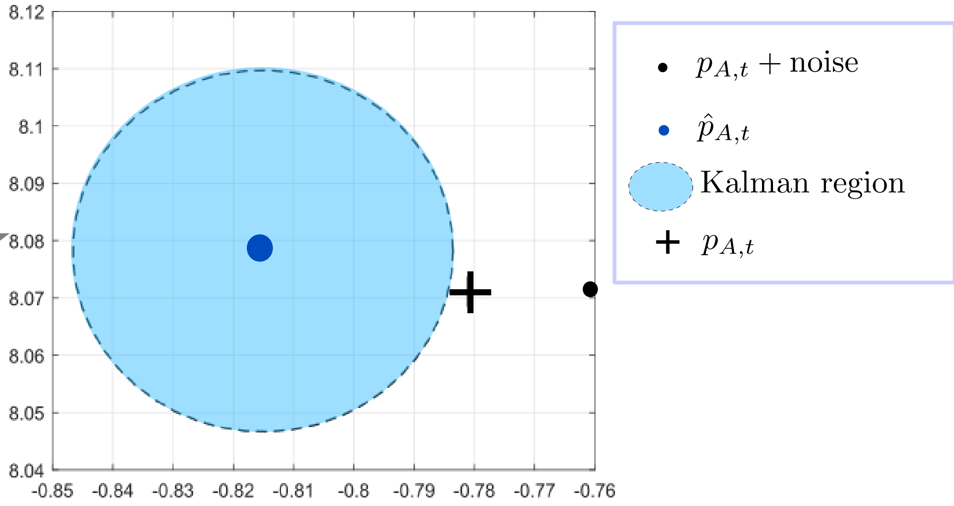
LET US ZOOM IN



ANOTHER RUN
Two robots run their **return-to-base** program

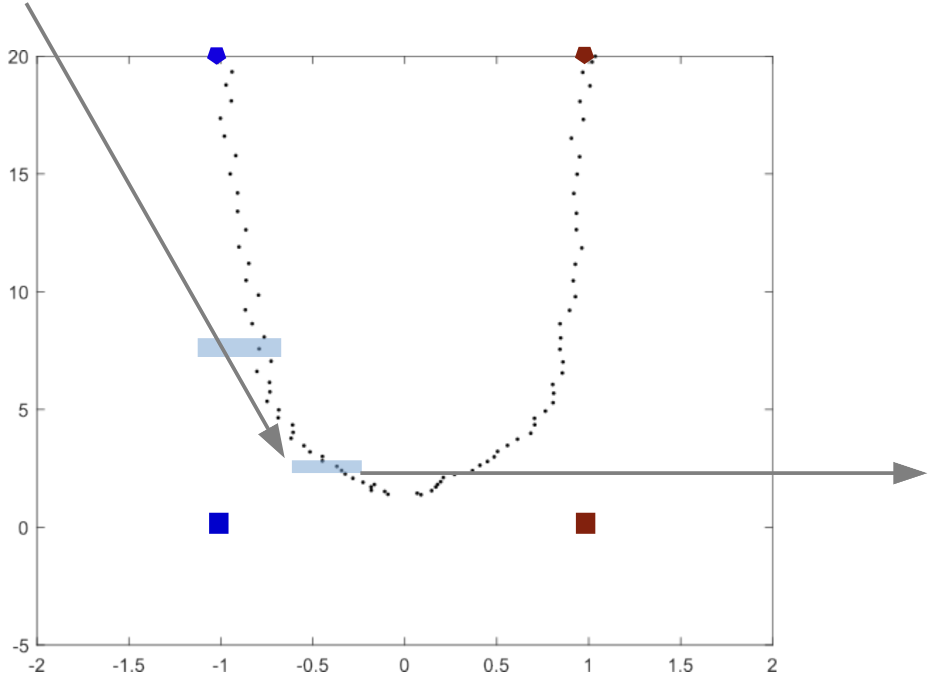
LET US ZOOM IN

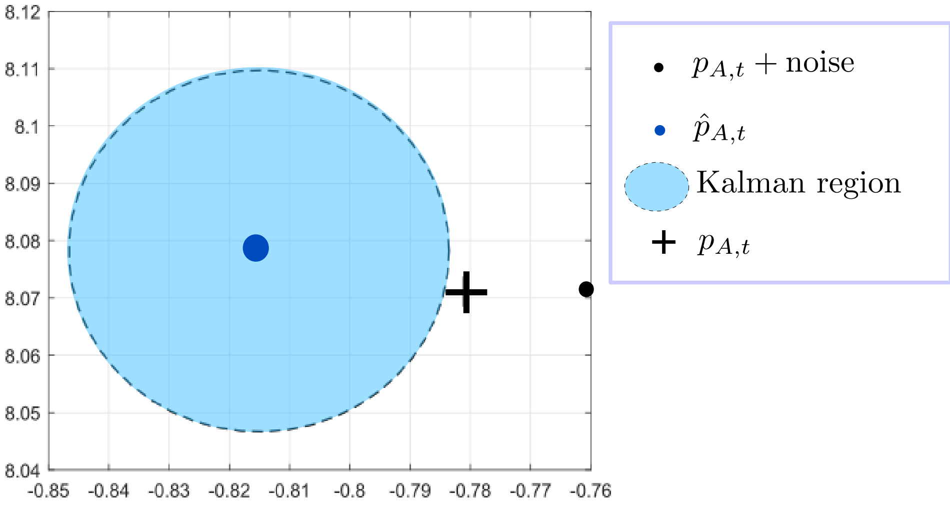


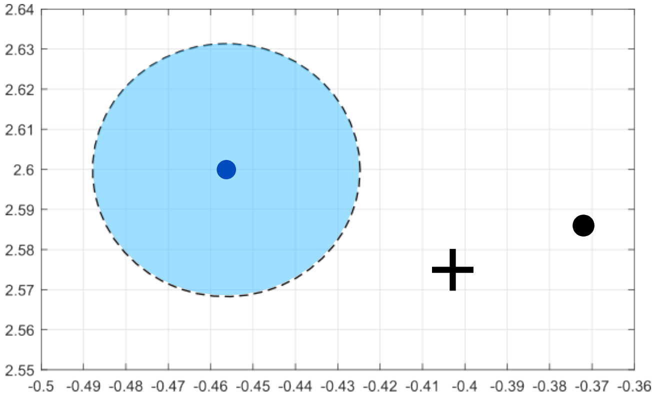


ANOTHER RUN
Two robots run their **return-to-base** program

LET US ZOOM IN

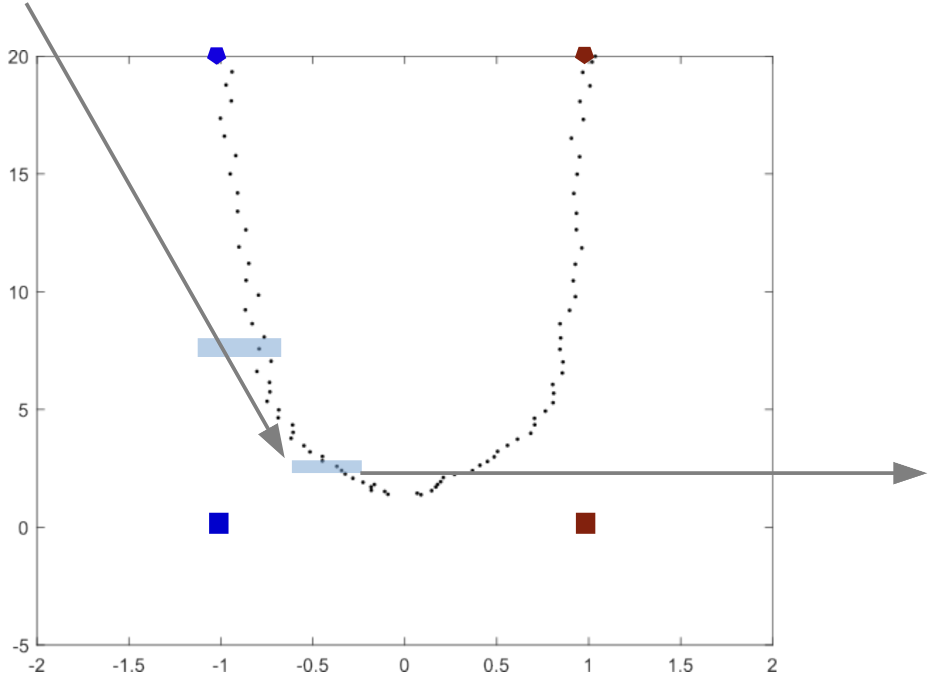


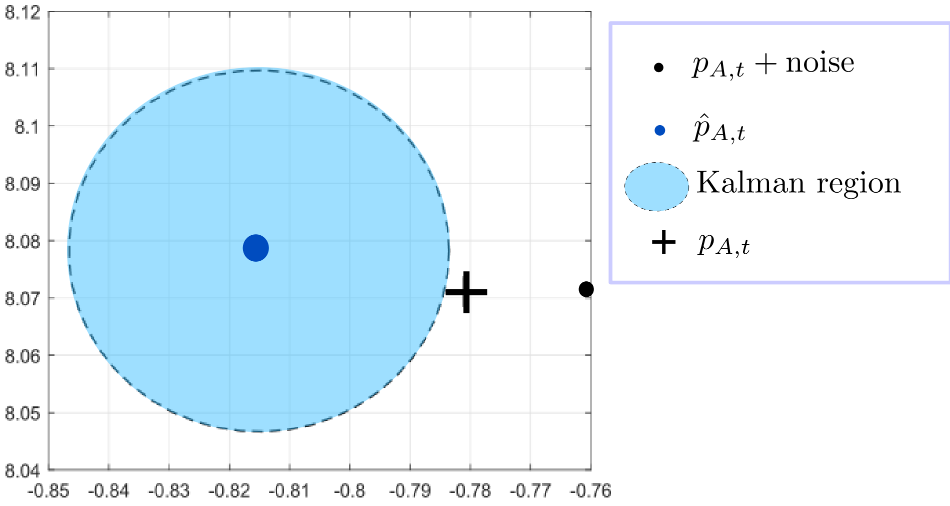


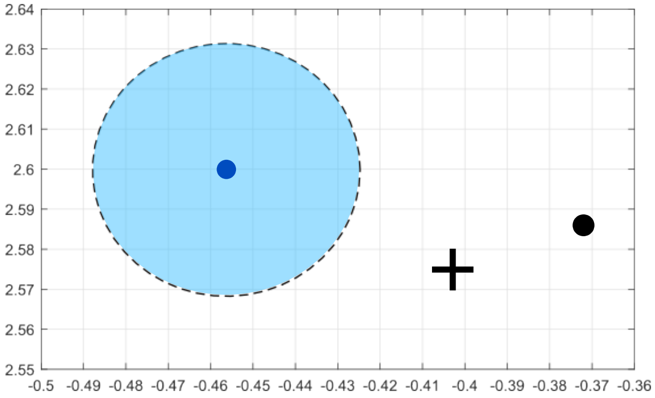


ANOTHER RUN
Two robots run their **return-to-base** program

LET US ZOOM IN







SCALAR STATIC CASE

$$Y = X + W$$

SCALAR STATIC CASE

$$Y = X + W$$

$$X \sim \mathcal{N}(0, \sigma_x^2) \qquad W \sim \mathcal{N}(0, \sigma_w^2)$$

SCALAR STATIC CASE

$$Y = X + W$$

$$X \sim \mathcal{N}(0, \sigma_x^2) \qquad W \sim \mathcal{N}(0, \sigma_w^2)$$

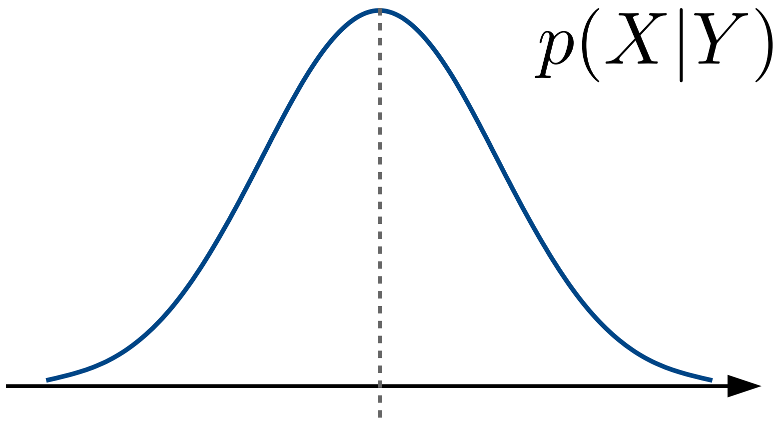
KF recipe:

SCALAR STATIC CASE

$$Y = X + W$$

$$X \sim \mathcal{N}(0, \sigma_x^2) \qquad W \sim \mathcal{N}(0, \sigma_w^2)$$

KF recipe:

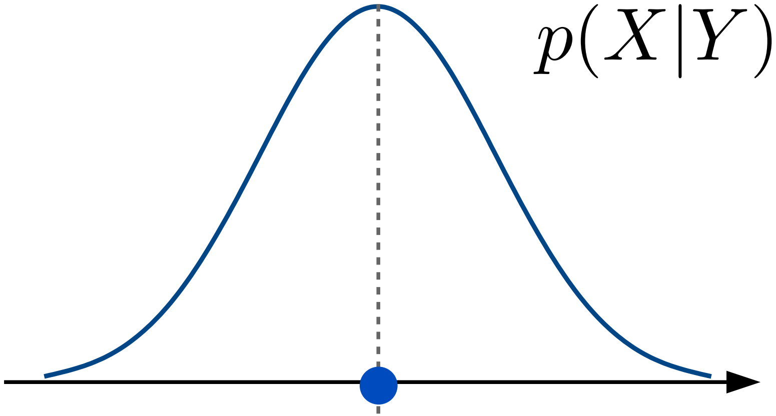


SCALAR STATIC CASE

$$Y = X + W$$

$$X \sim \mathcal{N}(0, \sigma_x^2) \qquad W \sim \mathcal{N}(0, \sigma_w^2)$$

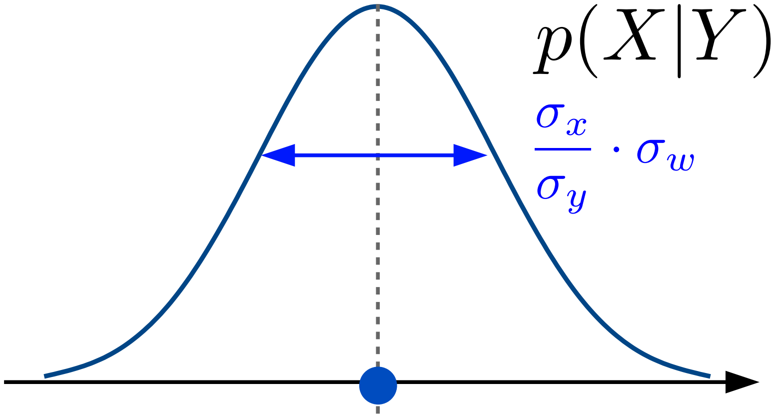
KF recipe:



SCALAR STATIC CASE

$$Y = X + W, \quad X \sim \mathcal{N}(0, \sigma_x^2), \quad W \sim \mathcal{N}(0, \sigma_w^2)$$

KF recipe:

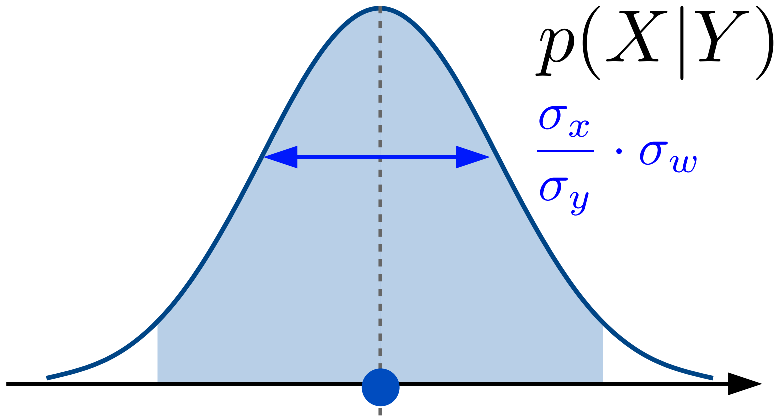


$$\hat{X}\left(= \frac{\sigma_x^2}{\sigma_y^2} Y\right)$$

SCALAR STATIC CASE

$$Y = \underset{\mathcal{N}(0,\sigma_x^2)}{\underset{\wr}{X}} + W \sim \mathcal{N}(0,\sigma_w^2)$$

KF recipe:



$p(X|Y)$

$\frac{\sigma_x}{\sigma_y} \cdot \sigma_w$

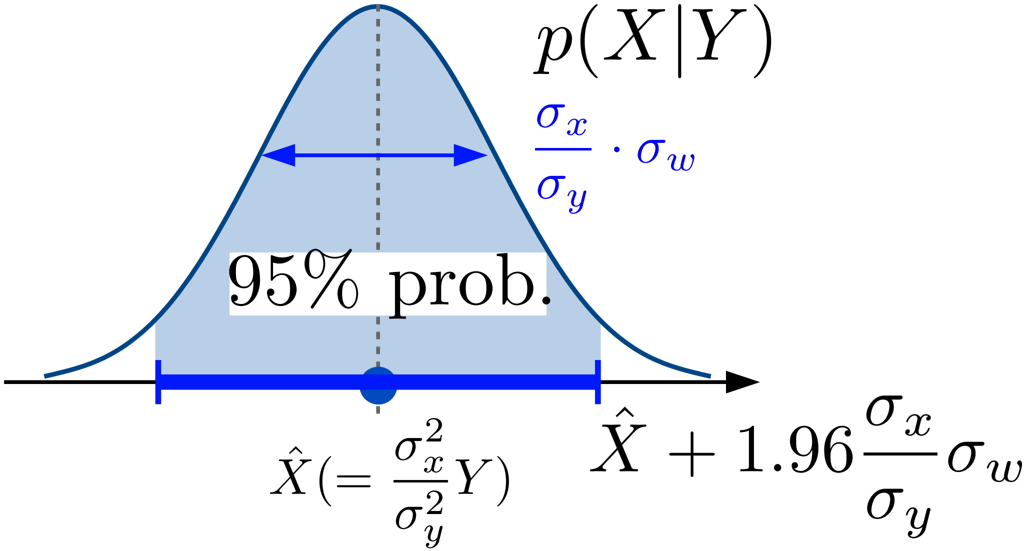
$\hat{X}(= \frac{\sigma_x^2}{\sigma_y^2} Y)$

SCALAR STATIC CASE

$$Y = X + W$$

$$X \sim \mathcal{N}(0, \sigma_x^2) \qquad W \sim \mathcal{N}(0, \sigma_w^2)$$

KF recipe:

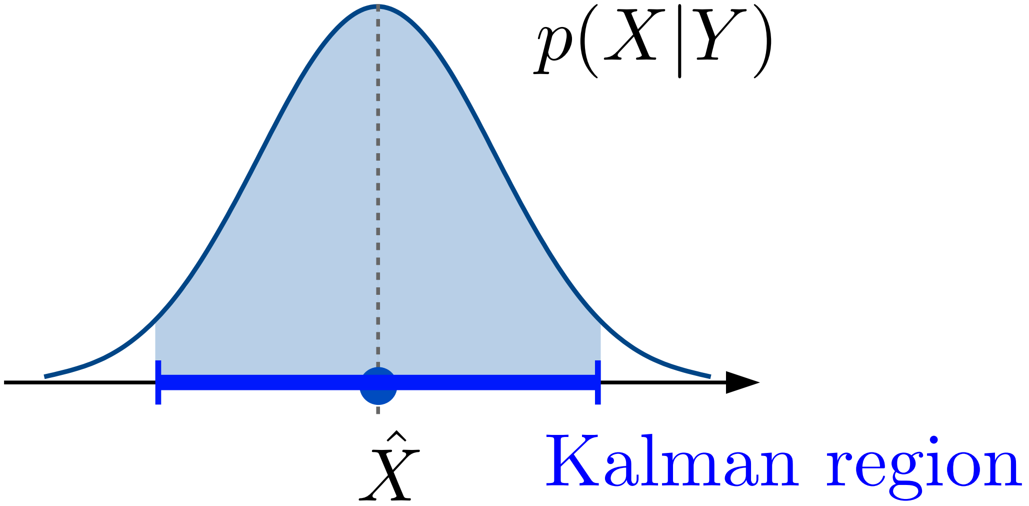


SCALAR STATIC CASE

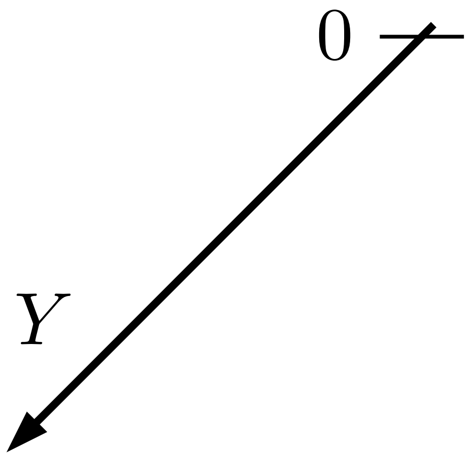
$$Y = X + W$$

$X \sim \mathcal{N}(0, \sigma_x^2)$, $W \sim \mathcal{N}(0, \sigma_w^2)$

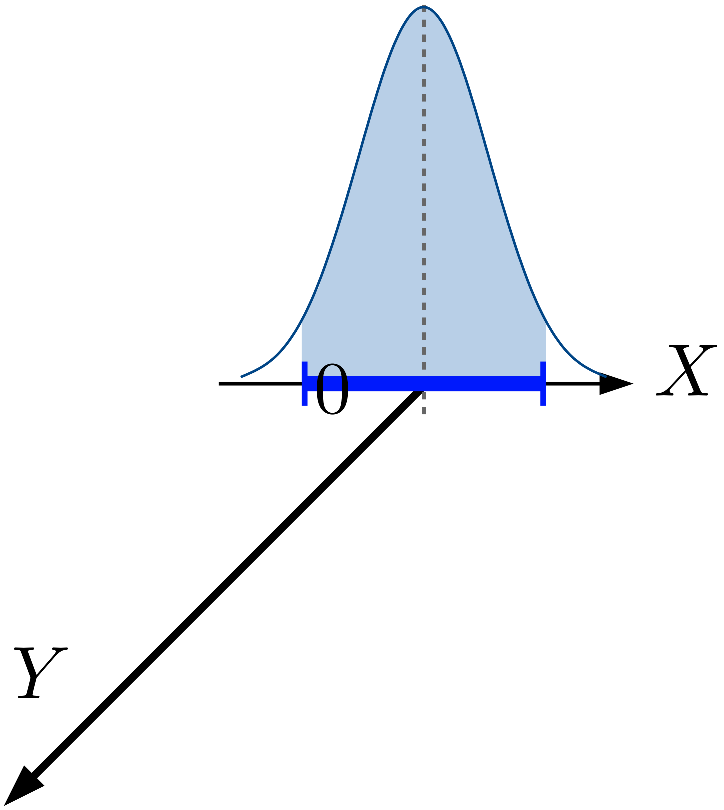
KF recipe:



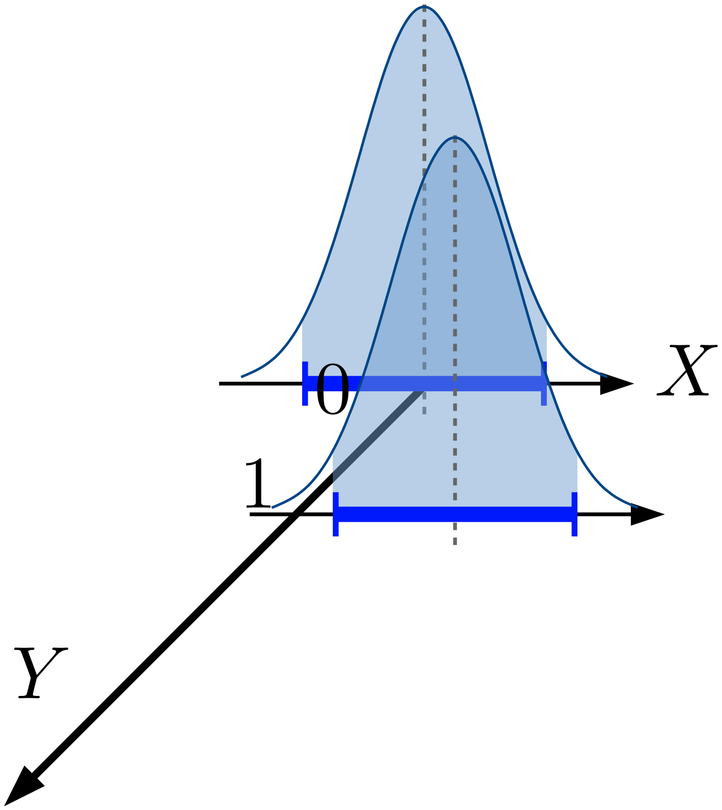
SCALAR STATIC CASE $Y = X + W$



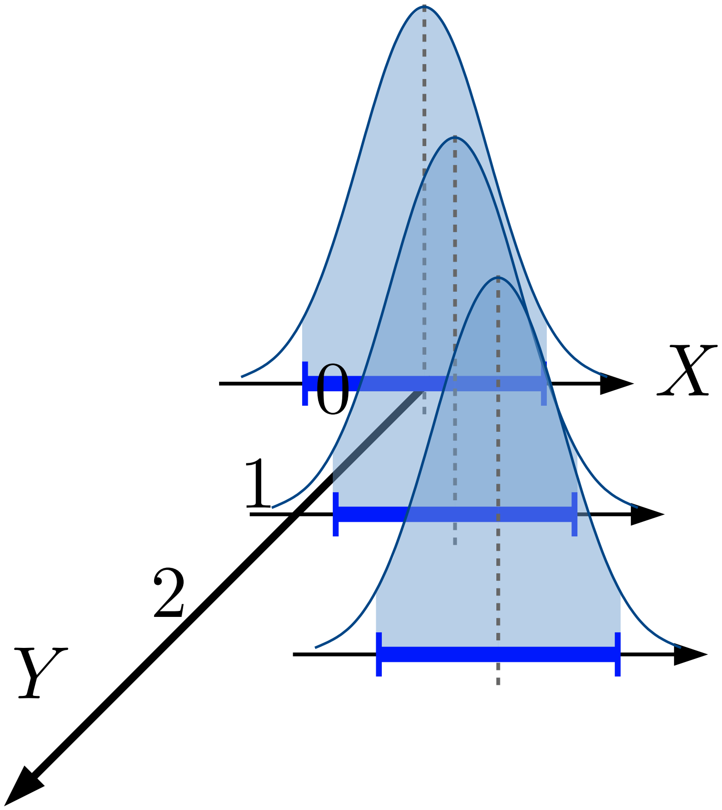
SCALAR STATIC CASE $Y = X + W$



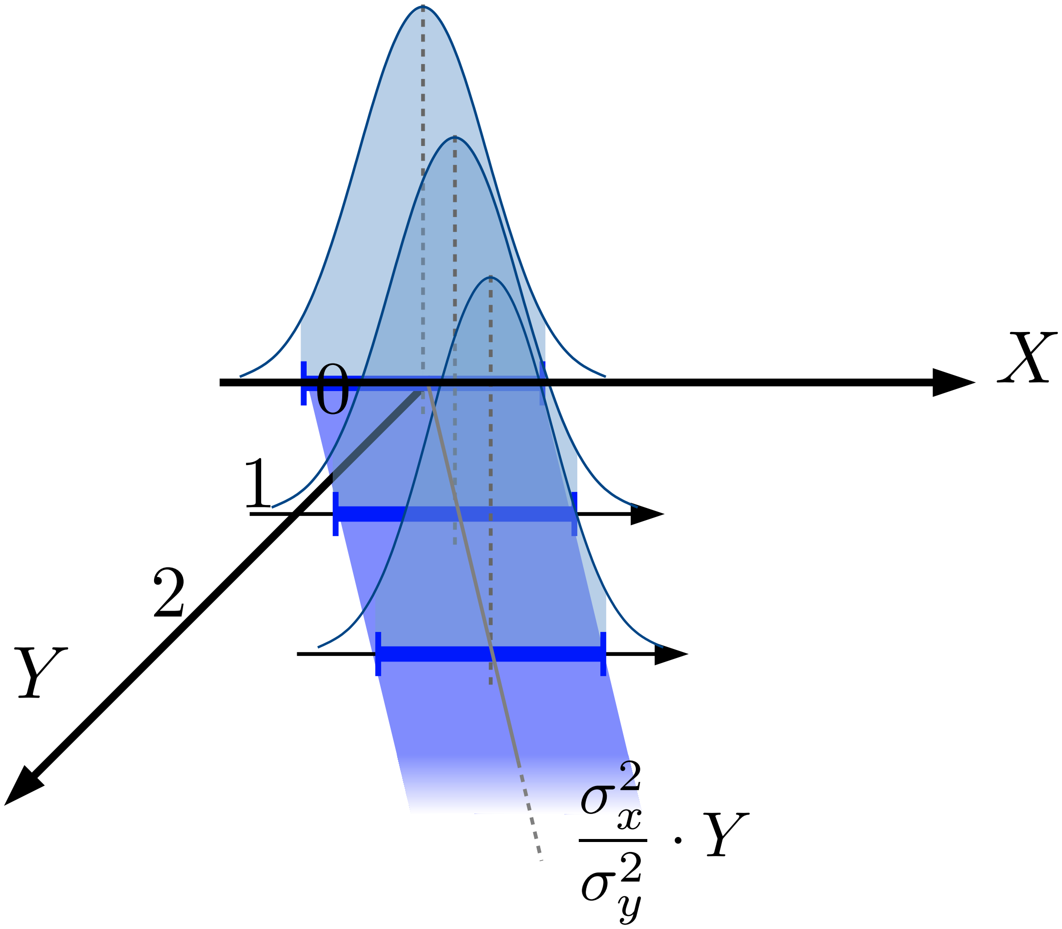
SCALAR STATIC CASE $Y = X + W$



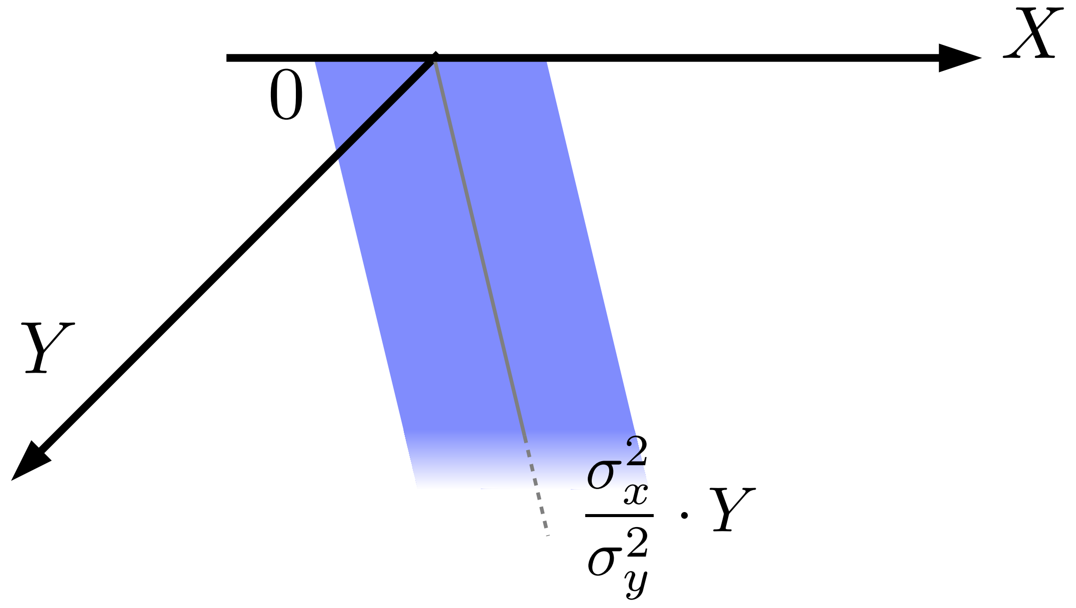
SCALAR STATIC CASE $Y = X + W$



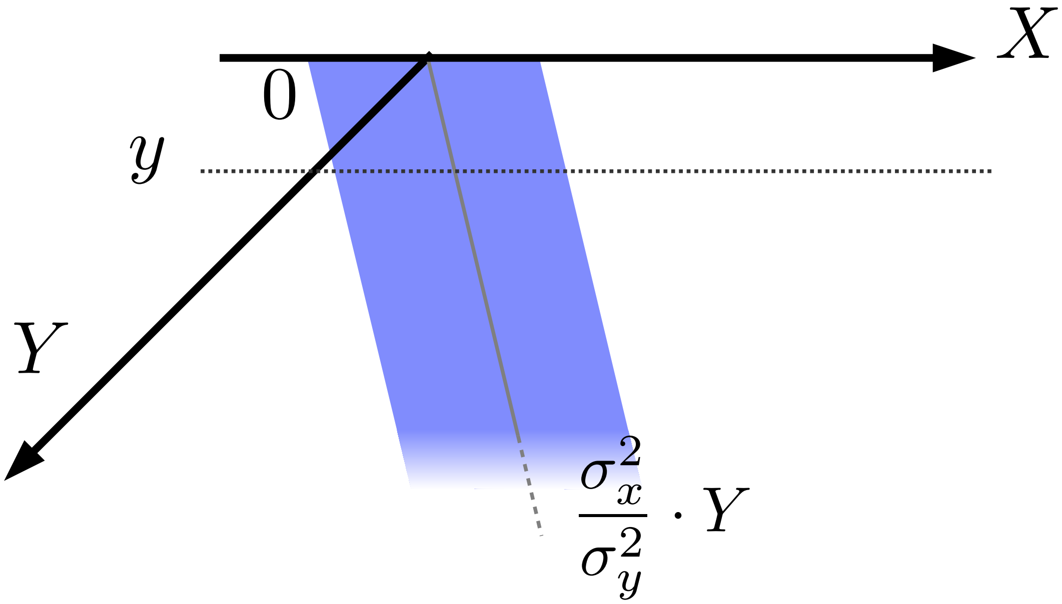
SCALAR STATIC CASE $Y = X + W$



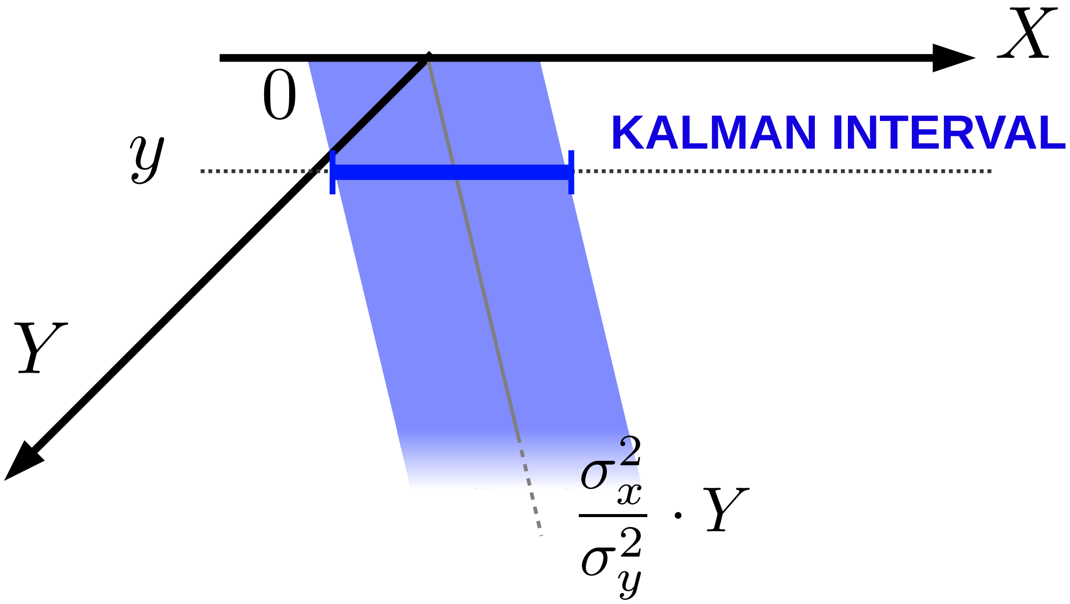
SCALAR STATIC CASE $Y = X + W$



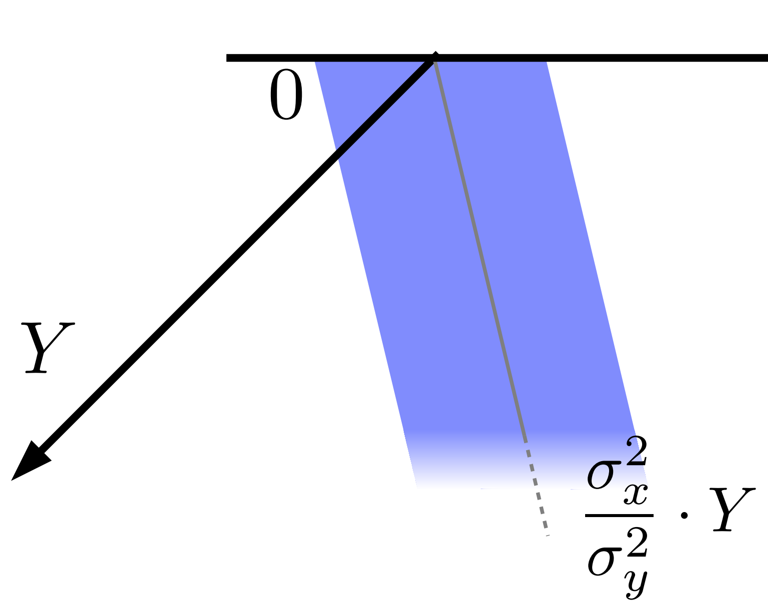
SCALAR STATIC CASE $Y = X + W$



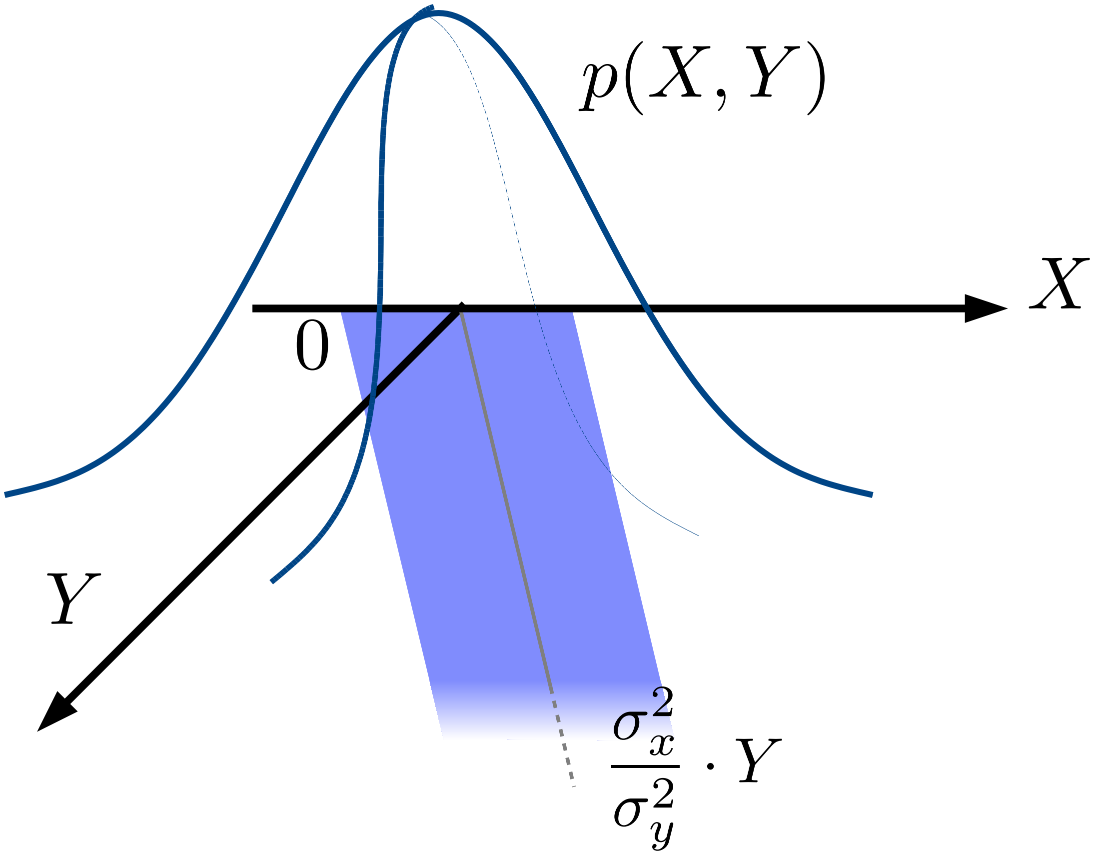
SCALAR STATIC CASE $Y = X + W$



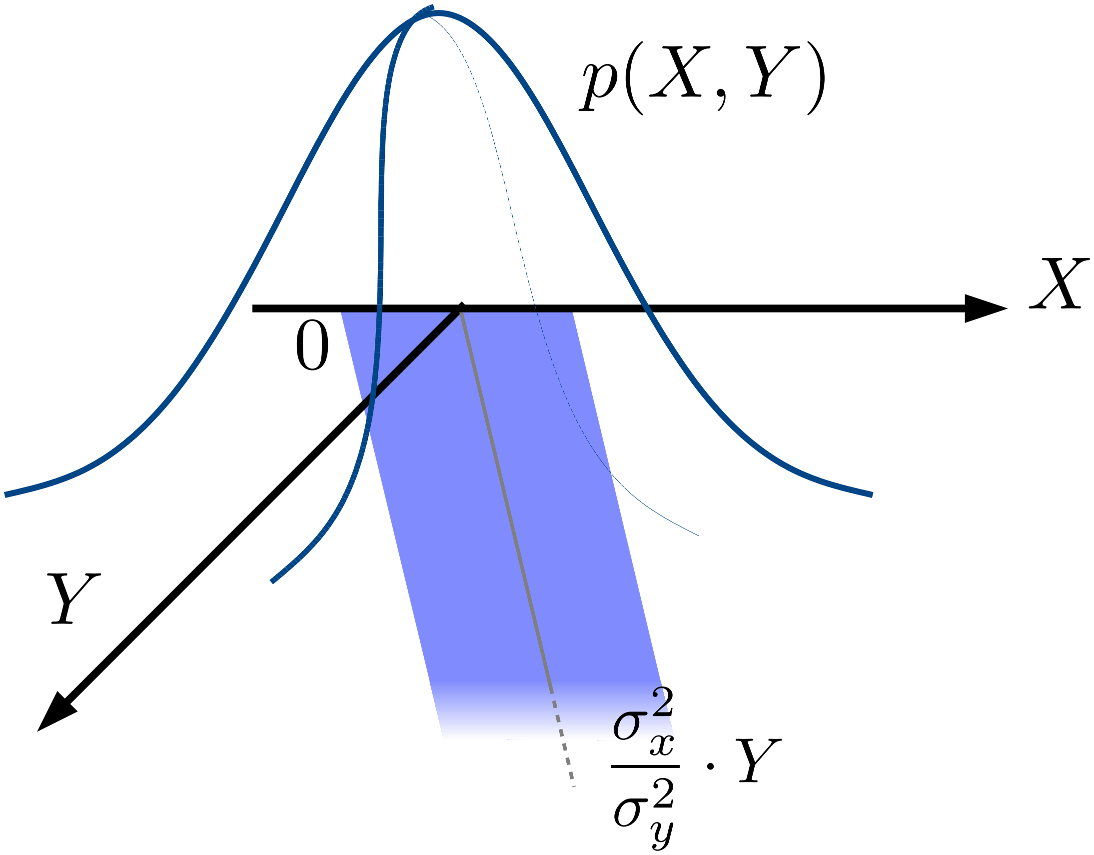
SCALAR STATIC CASE $Y = X + W$



SCALAR STATIC CASE $Y = X + W$

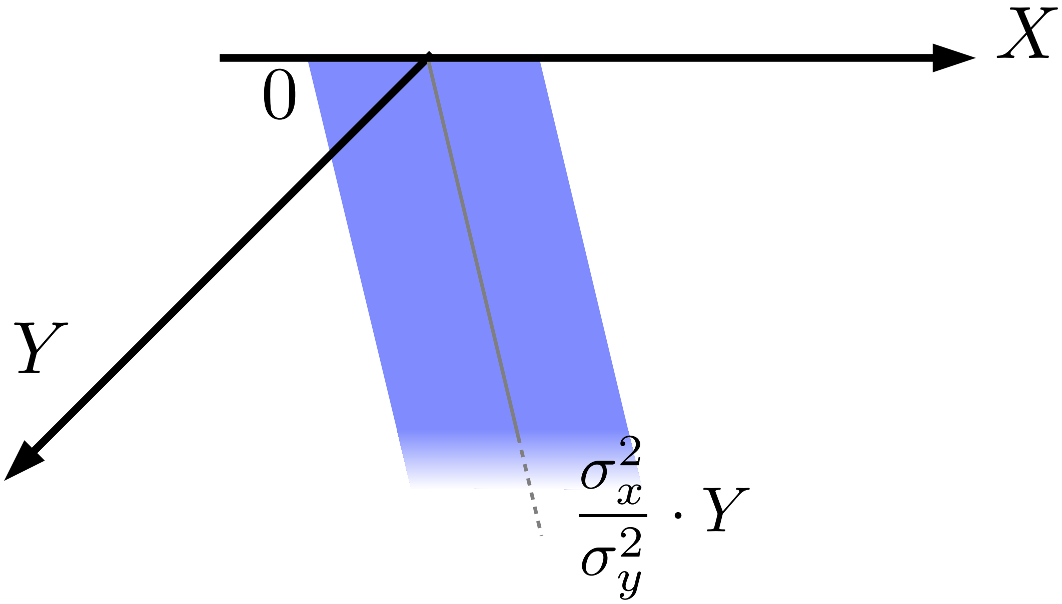


SCALAR STATIC CASE $\quad Y = X + W$

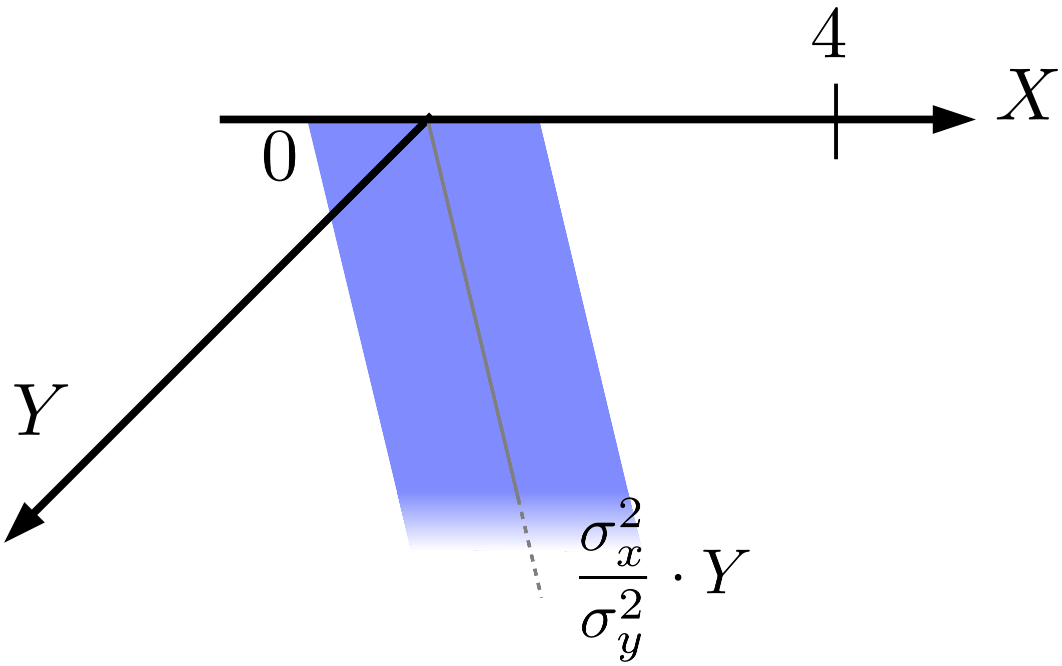


$Prob\{(X,Y) \in \text{BLUE REGION}\}$
$= 95\%$

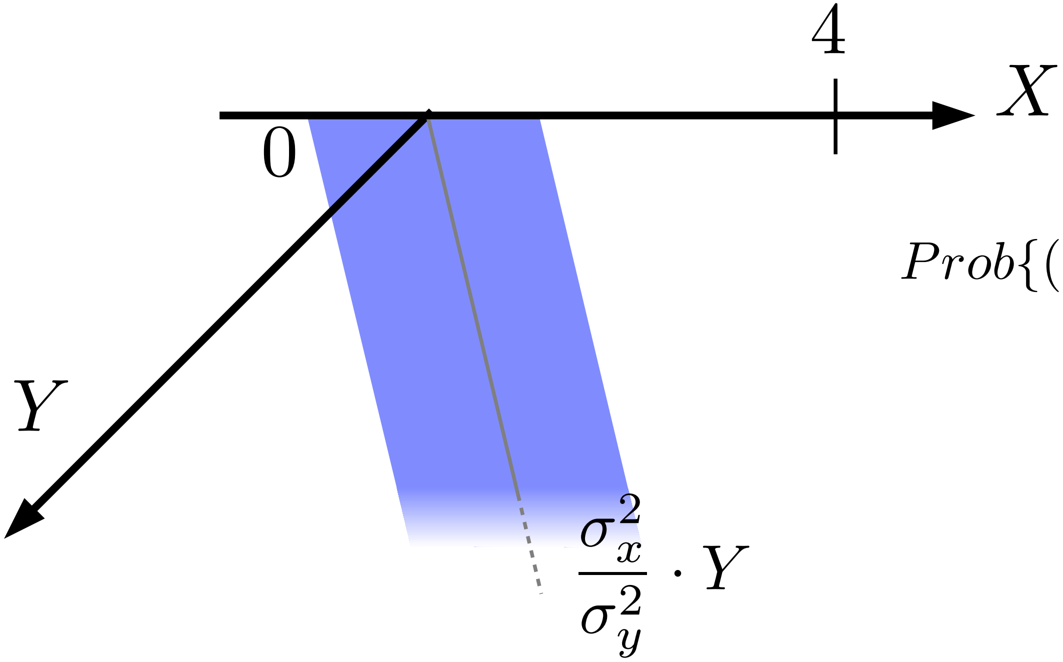
SCALAR STATIC CASE $Y = X + W$



SCALAR STATIC CASE $Y = X + W$

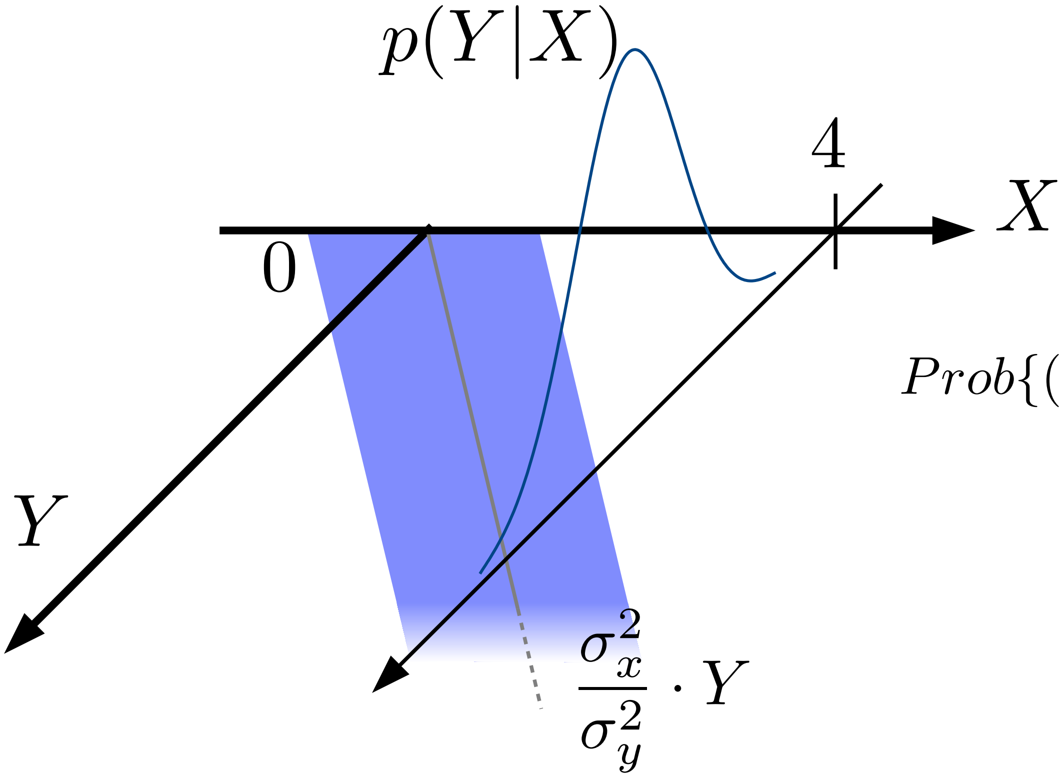


SCALAR STATIC CASE $\quad Y = X + W$



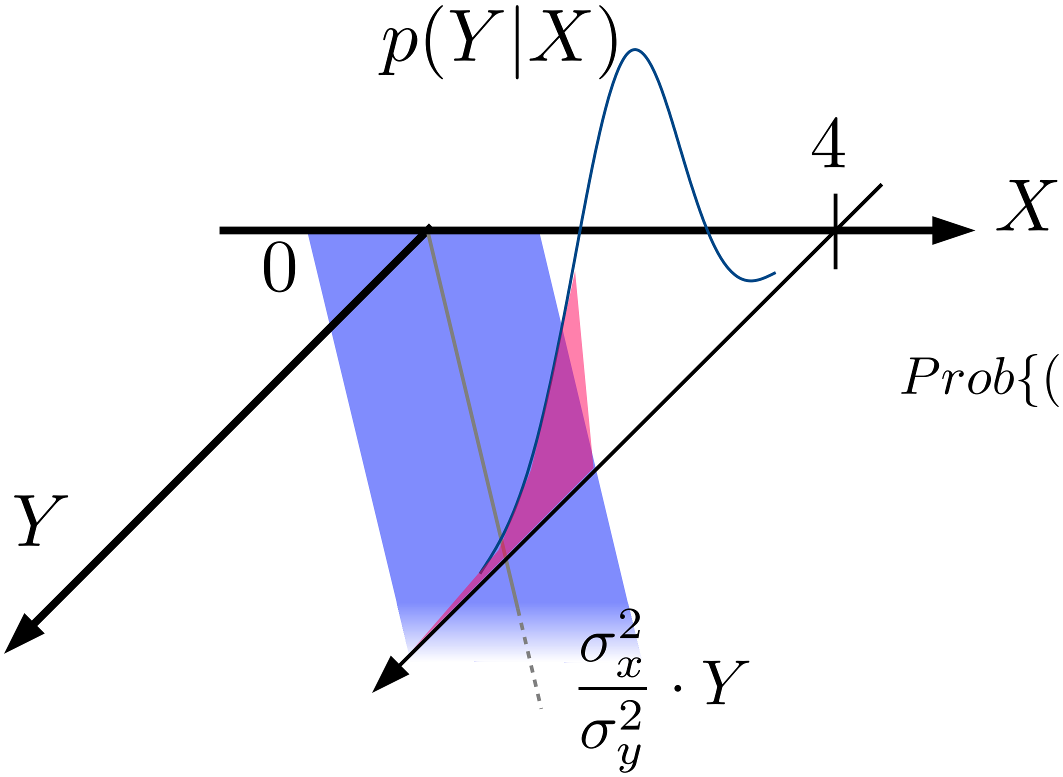
$$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$$
$$= \dots$$

SCALAR STATIC CASE $Y = X + W$



$$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$$
$$= \ldots$$

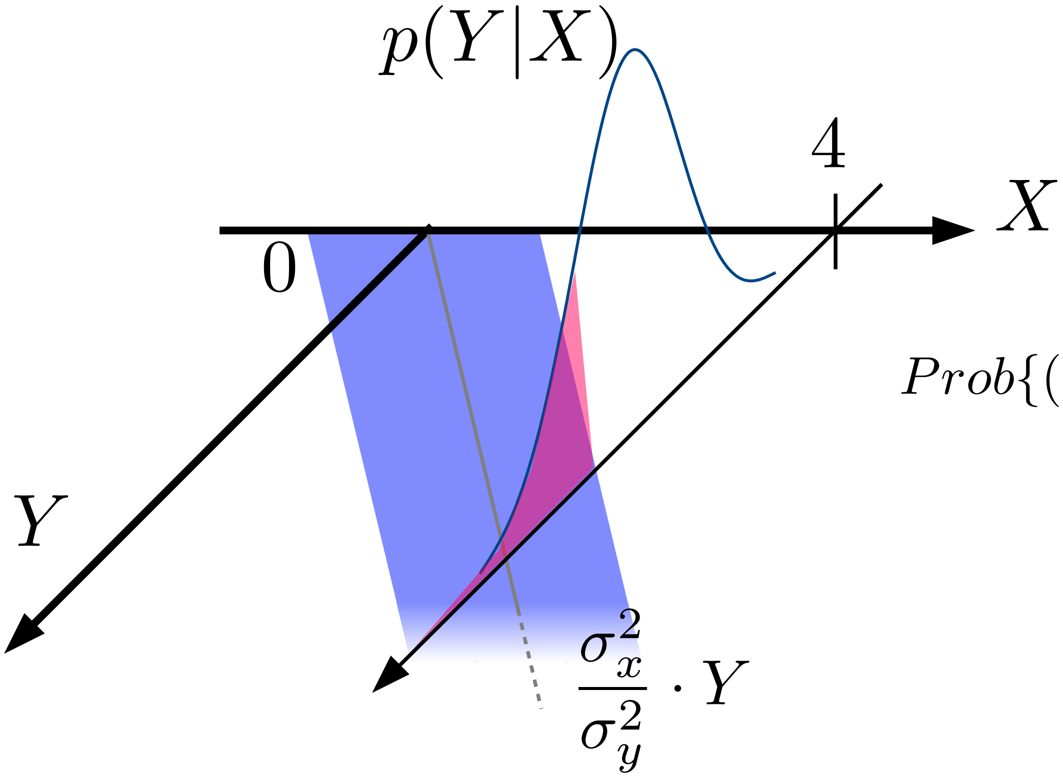
SCALAR STATIC CASE $\quad Y = X + W$



$$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$$
$$= \ldots$$

SCALAR STATIC CASE $Y = X + W$

$\sigma_x = 1, \sigma_w = 1$

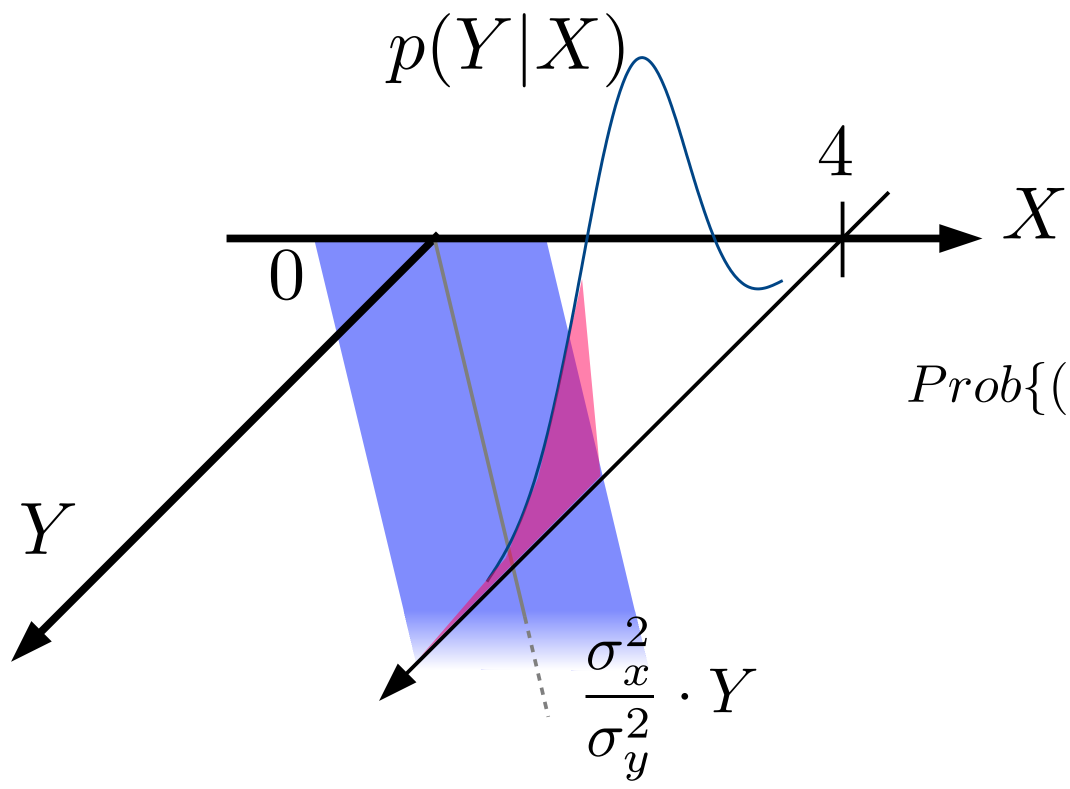


$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$

$= \ldots$

SCALAR STATIC CASE $Y = X + W$

$\sigma_x = 1, \sigma_w = 1$

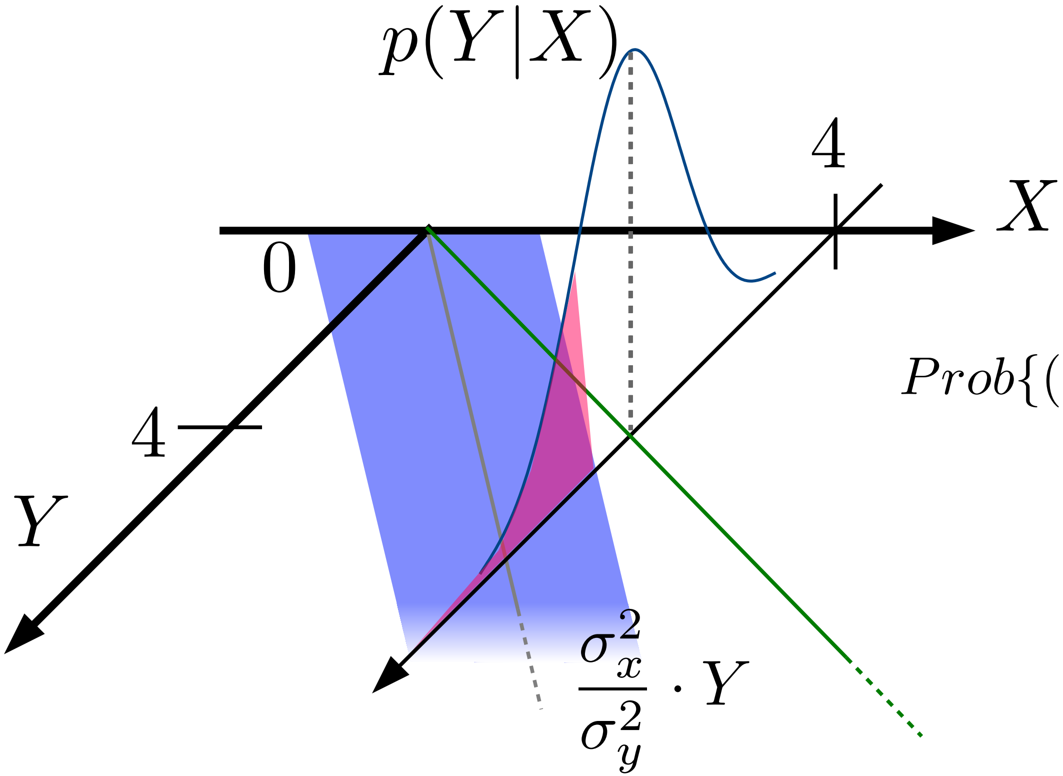


$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$

$\approx 11\%$

SCALAR STATIC CASE $Y = X + W$

$\sigma_x = 1, \sigma_w = 1$

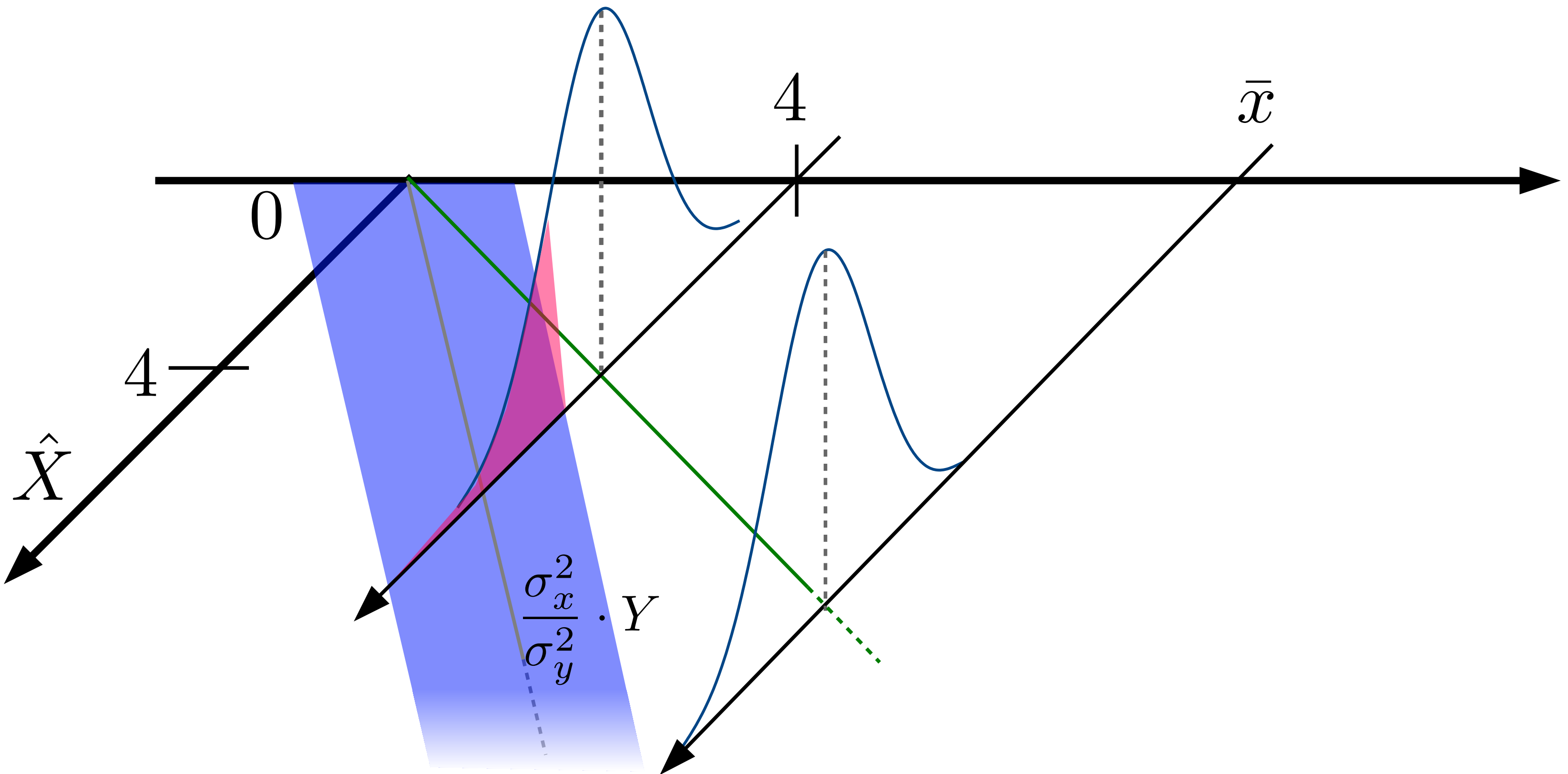


$Prob\{(X,Y) \in \text{BLUE REGION} | X = 4\}$

$\approx 11\%$

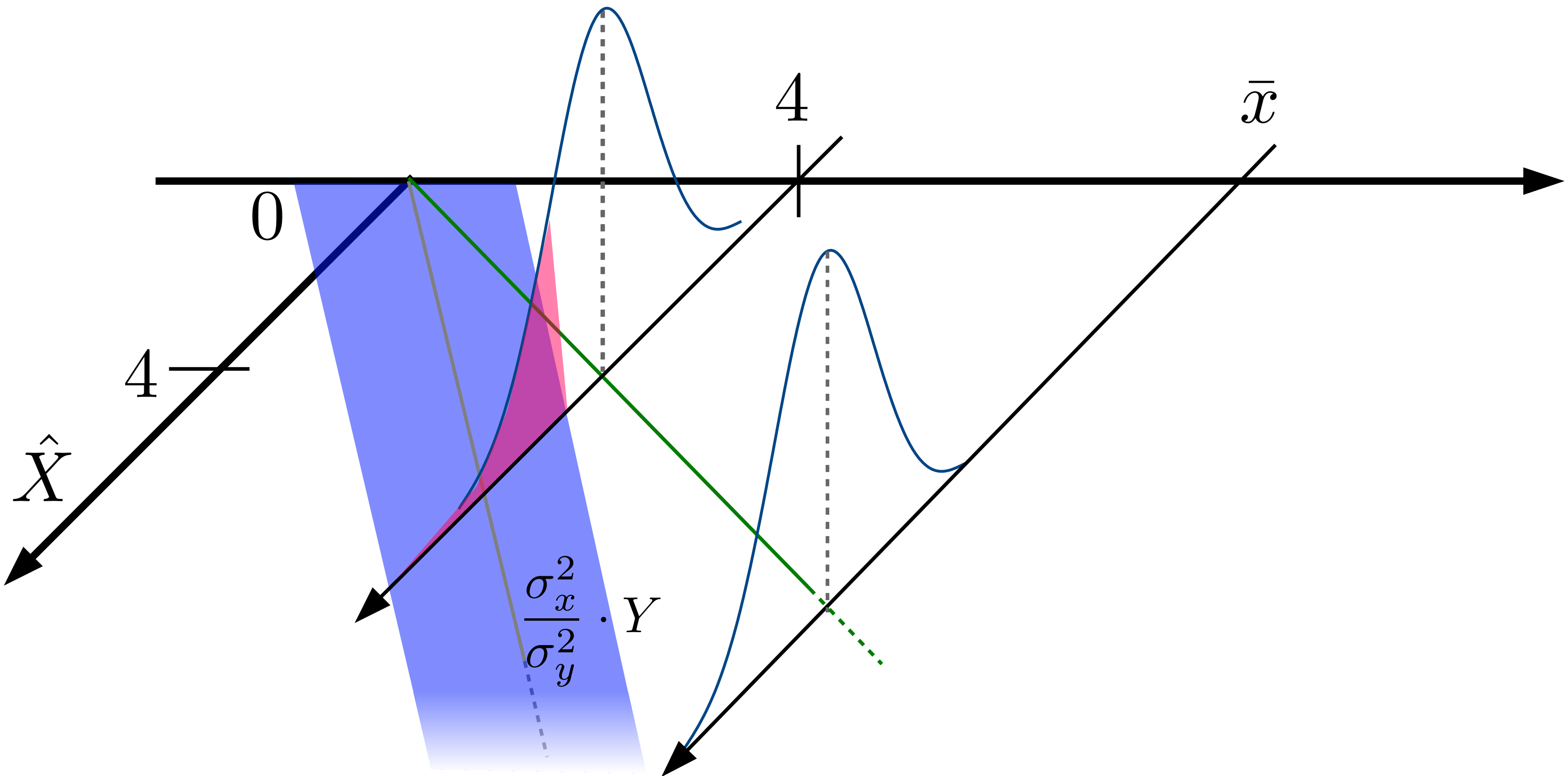
SCALAR STATIC CASE $\quad Y = X + W$

$$Prob\{(X,Y) \in \text{BLUE REGION} | X = \bar{x}\}$$



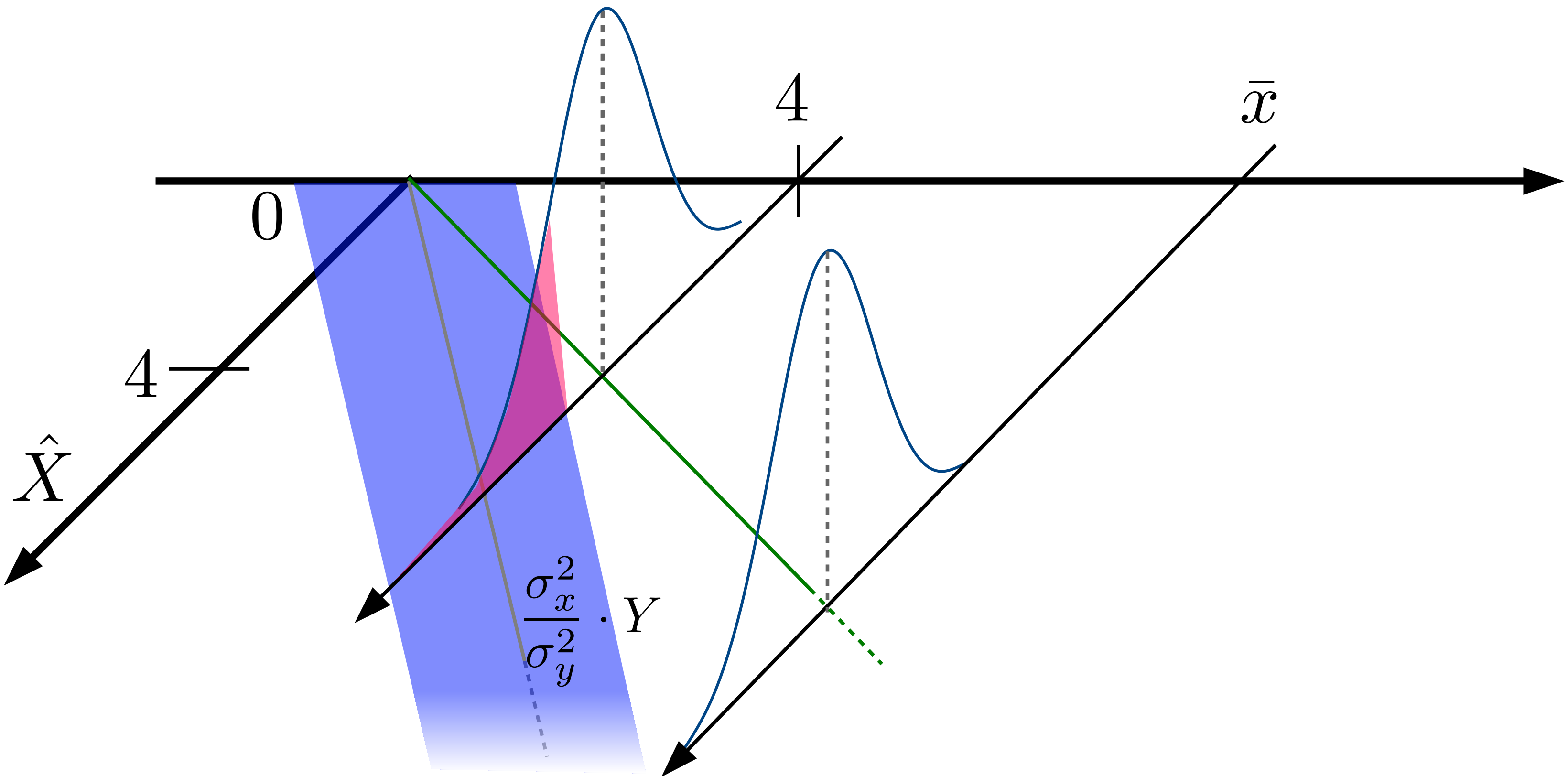
SCALAR STATIC CASE $\quad Y = X + W$

$$\lim_{\bar{x} \to \infty} Prob\{(X, Y) \in \text{BLUE REGION} | X = \bar{x}\}$$



SCALAR STATIC CASE $Y = X + W$

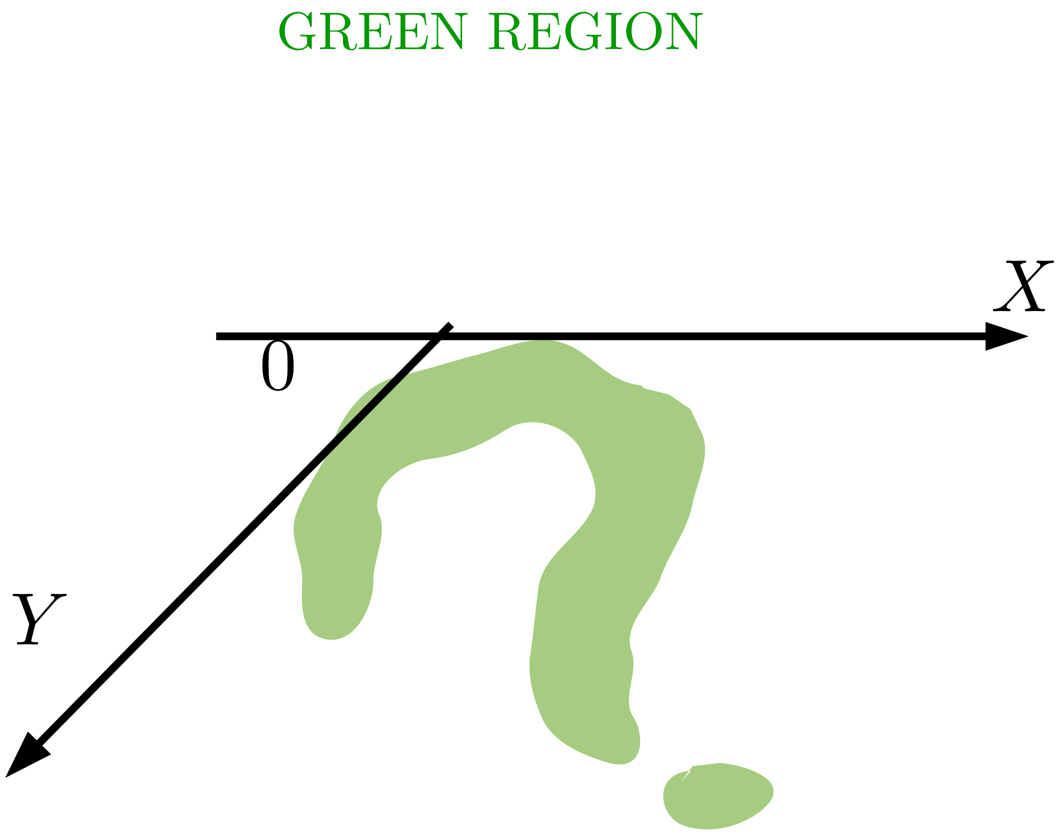
$$\lim_{\bar{x} \to \infty} Prob\{(X, Y) \in \text{BLUE REGION} | X = \bar{x}\} = 0$$



SCALAR STATIC CASE $Y = X + W$

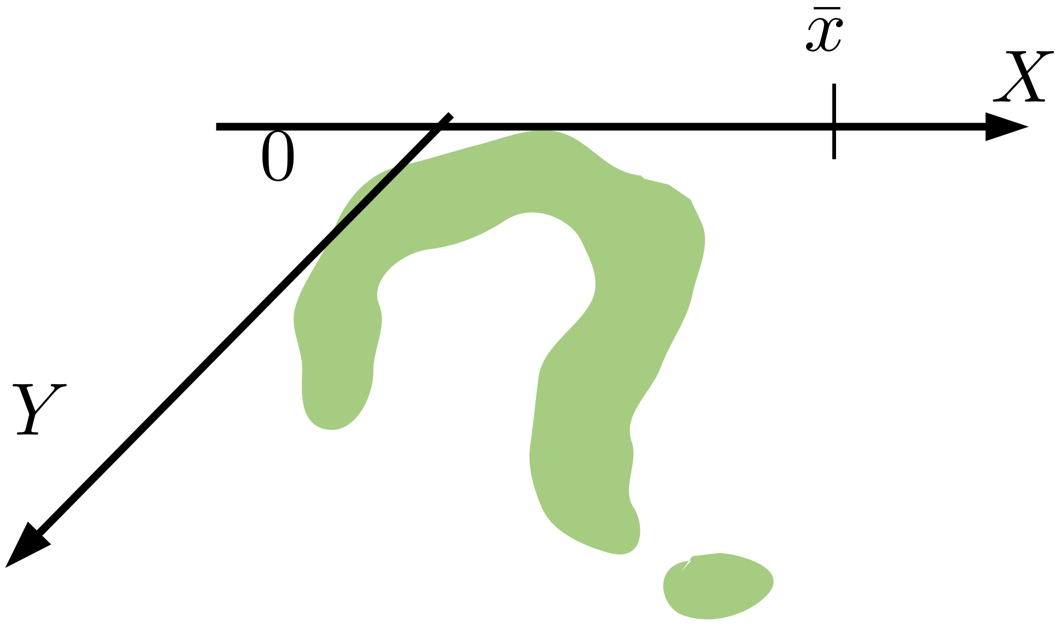
REGION

SCALAR STATIC CASE $Y = X + W$



SCALAR STATIC CASE $Y = X + W$

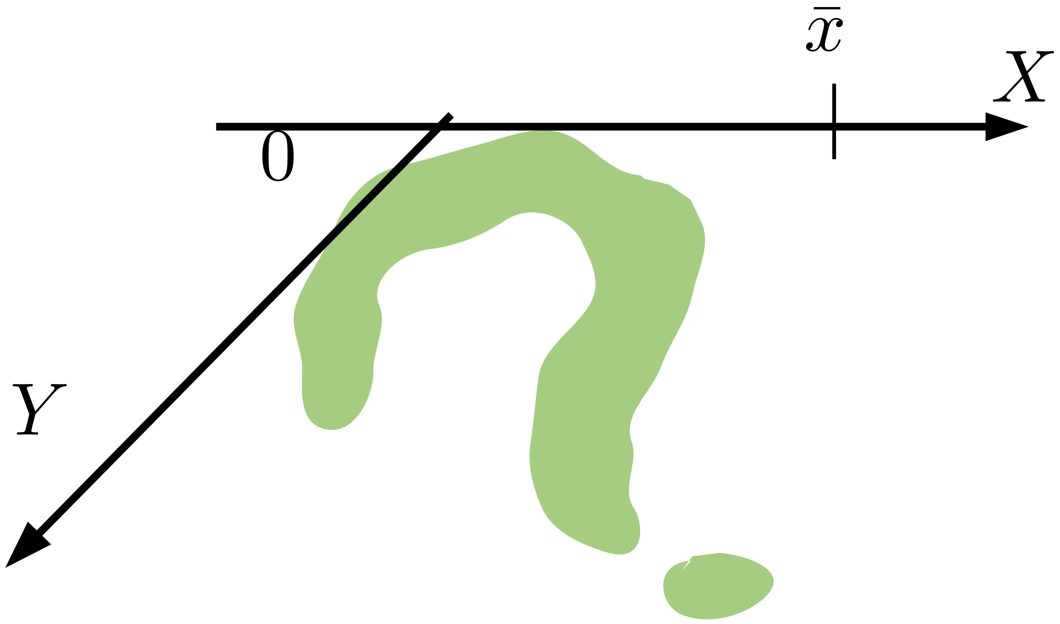
$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT$ for all $\bar{x}$



SCALAR STATIC CASE $Y = X + W$

"STATE CONDITIONAL PROPERTY"

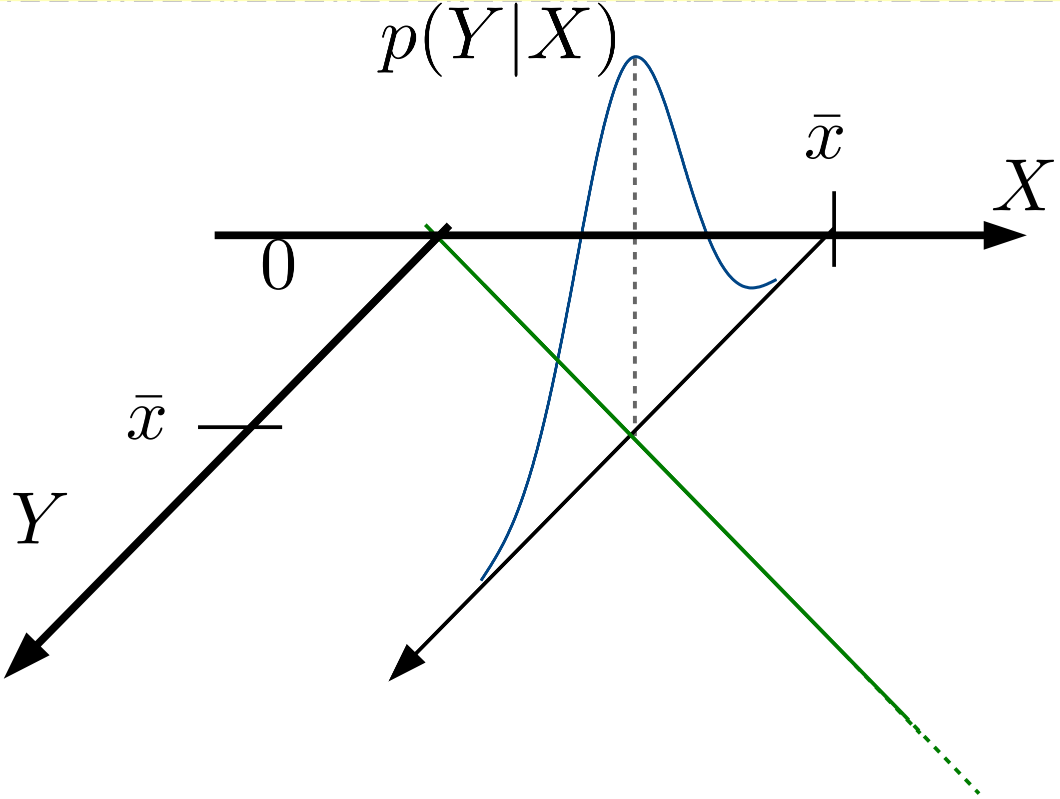
$$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT \text{ for all } \bar{x}$$



SCALAR STATIC CASE $\quad Y = X + W$

"STATE CONDITIONAL PROPERTY"

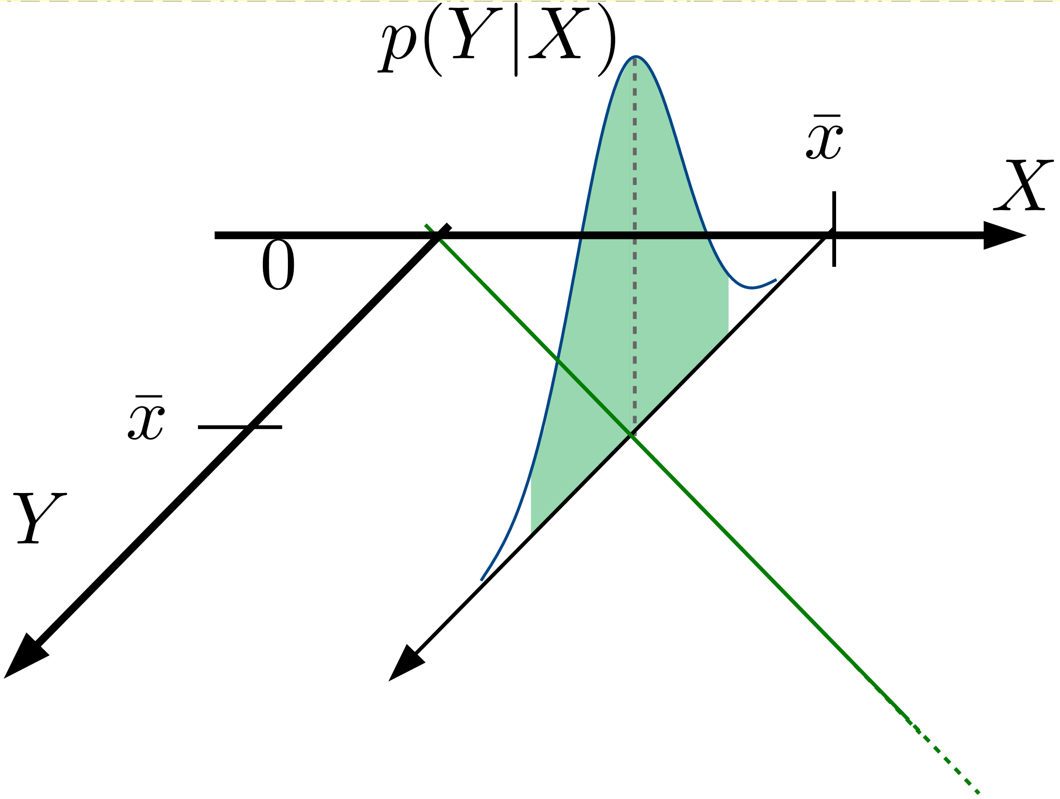
$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT$ for all $\bar{x}$



SCALAR STATIC CASE $Y = X + W$

"STATE CONDITIONAL PROPERTY"

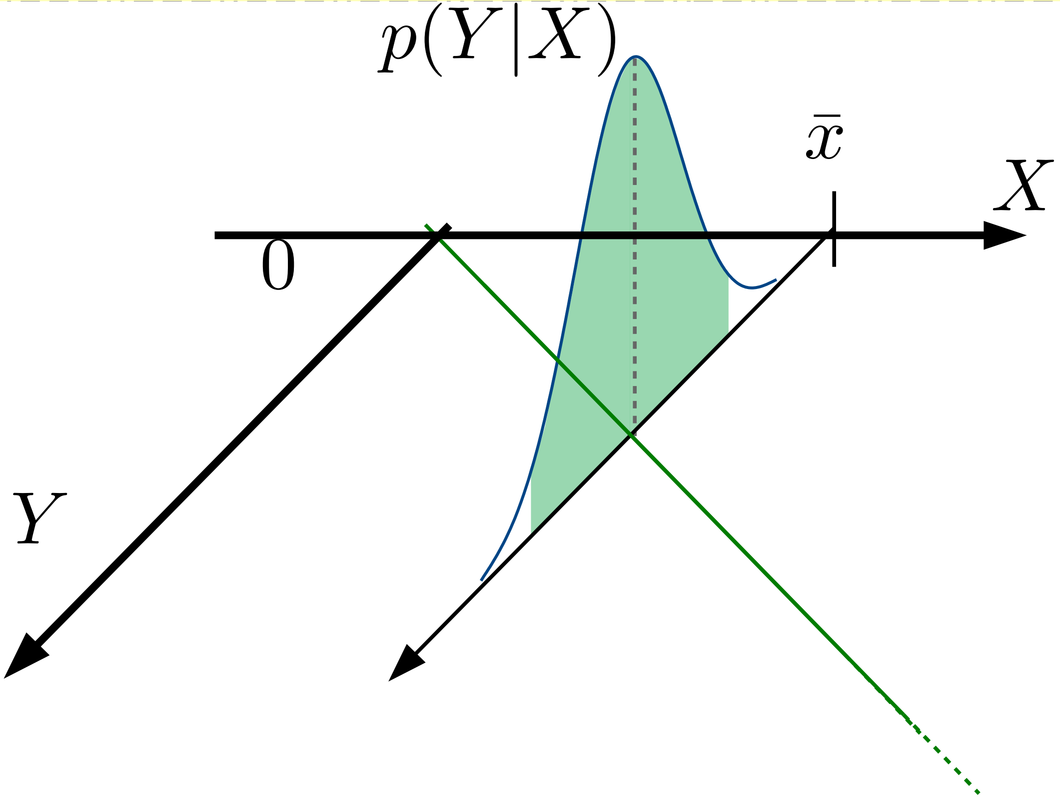
$$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT \text{ for all } \bar{x}$$



SCALAR STATIC CASE $Y = X + W$

"STATE CONDITIONAL PROPERTY"

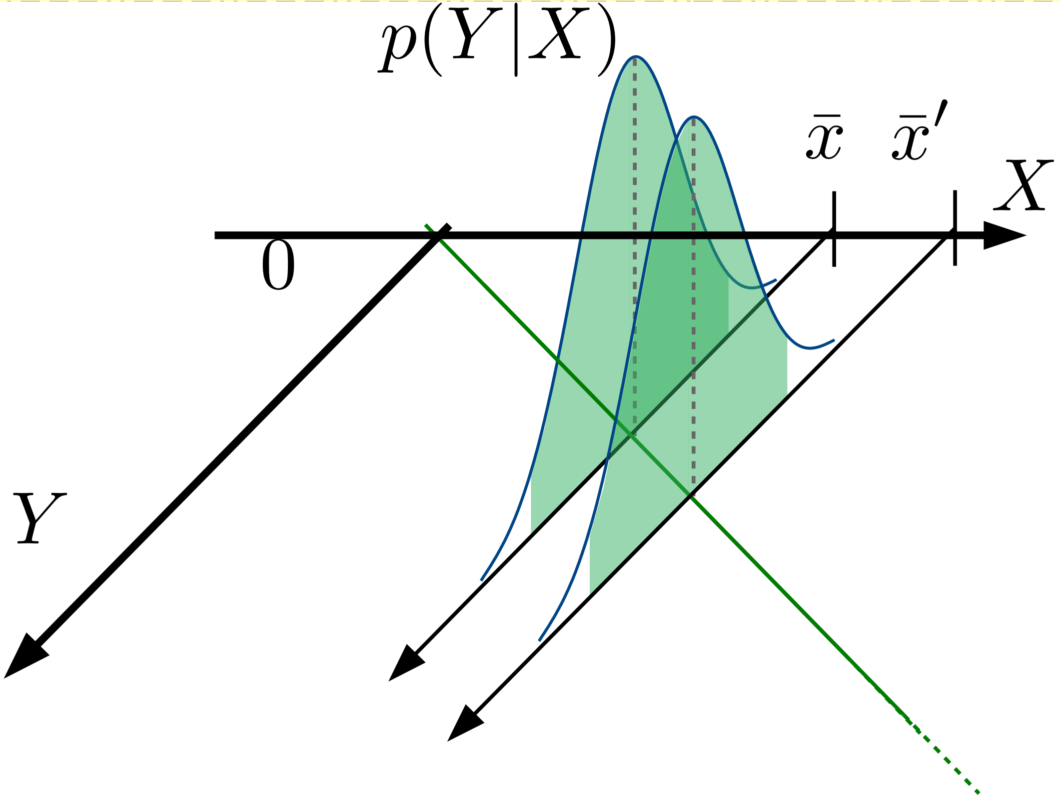
$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT$ for all $\bar{x}$



SCALAR STATIC CASE $Y = X + W$

"STATE CONDITIONAL PROPERTY"

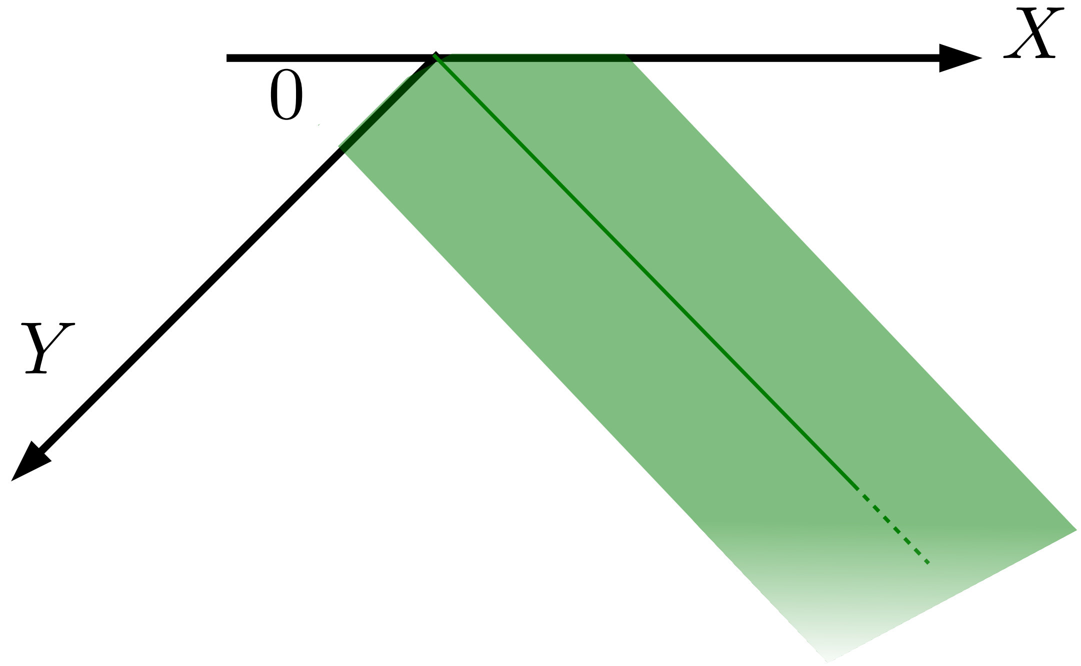
$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT$ for all $\bar{x}$



SCALAR STATIC CASE $Y = X + W$

"STATE CONDITIONAL PROPERTY"

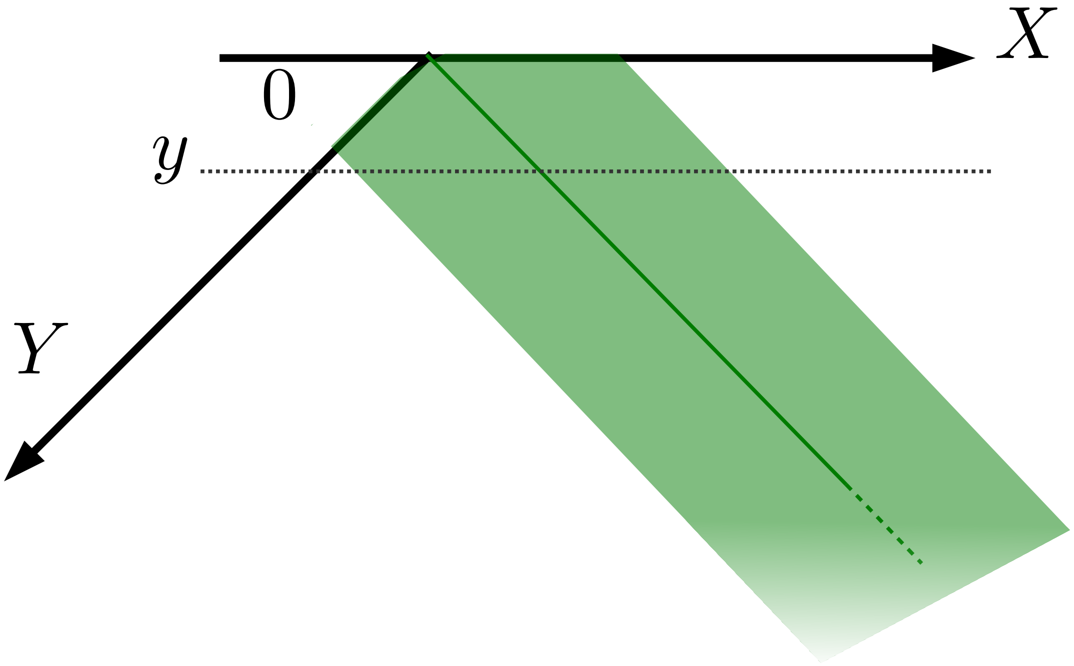
$$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT \text{ for all } \bar{x}$$



SCALAR STATIC CASE $\quad Y = X + W$

"STATE CONDITIONAL PROPERTY"

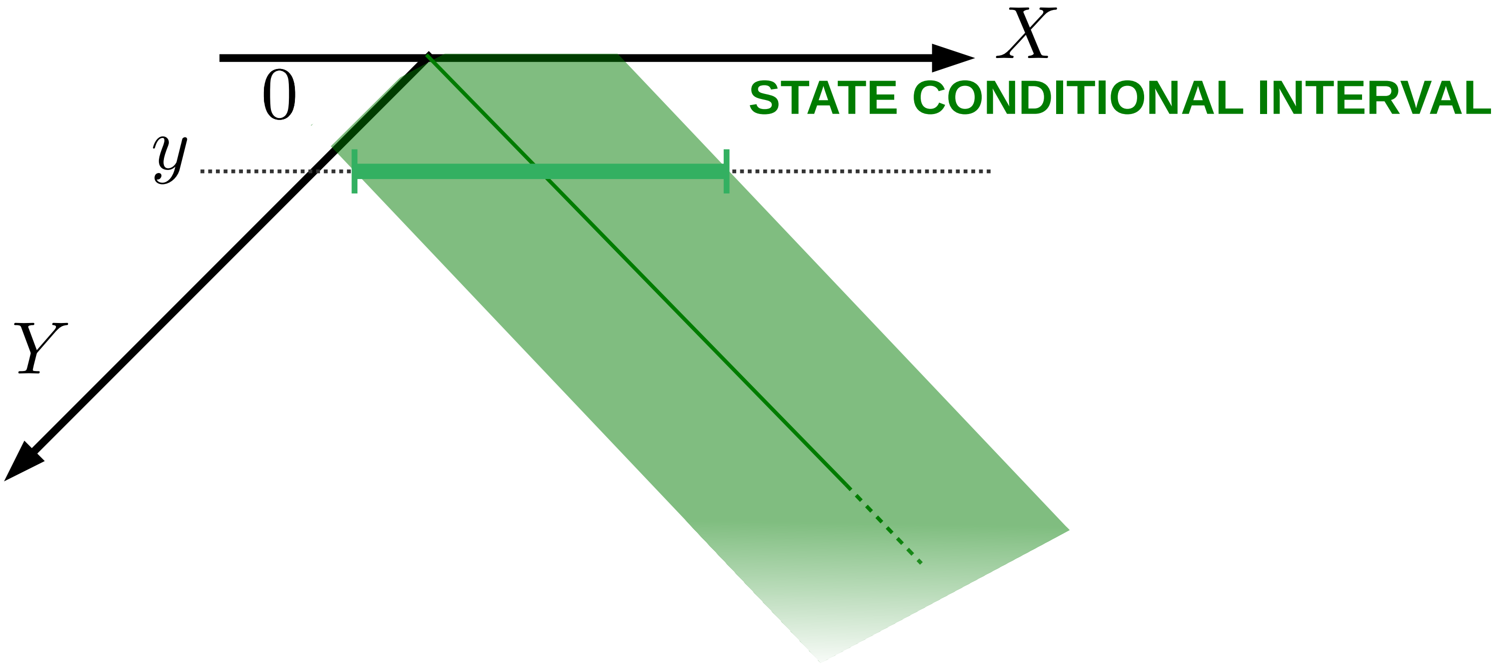
$$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT \text{ for all } \bar{x}$$



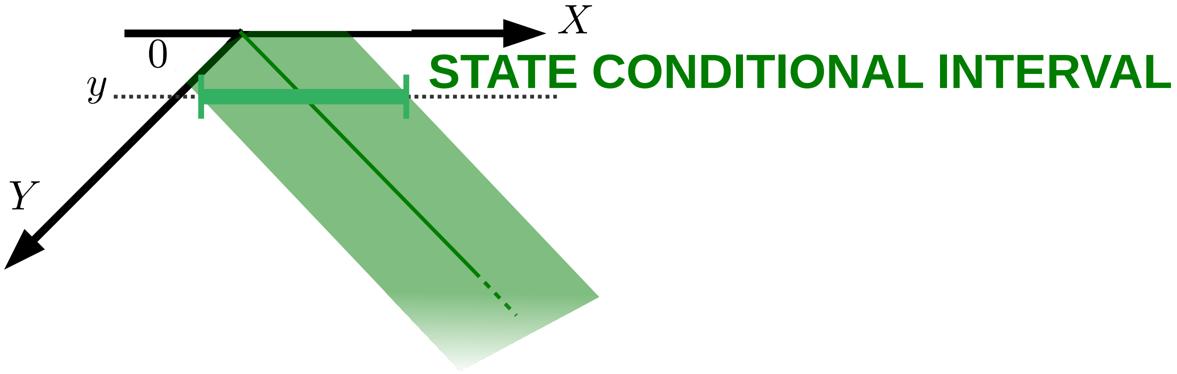
SCALAR STATIC CASE $Y = X + W$

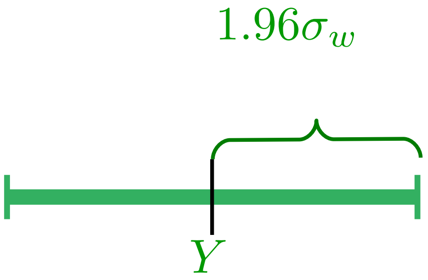
"STATE CONDITIONAL PROPERTY"

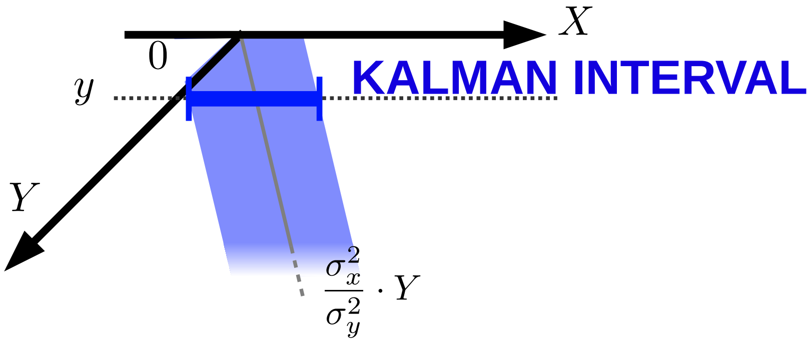
$$Prob\{(X,Y) \in \text{GREEN REGION} | X = \bar{x}\} = CONSTANT \text{ for all } \bar{x}$$

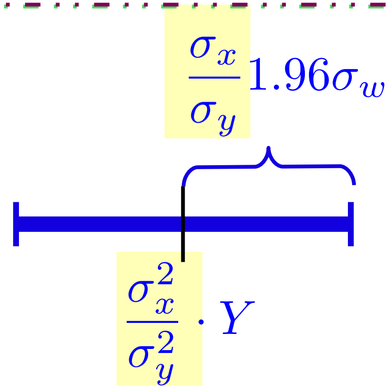


SCALAR STATIC CASE $Y = X + W$









DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

$$\hat{x}_t^{SCF}$$

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \{x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n)\},$$

$$\hat{x}_t^{SCF}$$

$$\Pi_t$$

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

KALMAN-RELATED QUANTITIES

DYNAMIC CASE

$$\begin{cases} x_{t+1} = F x_t + v_t \\ y_t = H x_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

- DATA-INDEPENDENT MATRIX

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

KALMAN-RELATED QUANTITIES

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

- DATA-INDEPENDENT MATRIX
- CAN BE COMPUTED RECURSIVELY

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

KALMAN-RELATED QUANTITIES

Pseudocode: SCF initialization

- Set the probability level $\alpha \in (0,1)$
- Let
  - ⋆ $U_0 A_0 \leftarrow 0$
  - ⋆ $\hat{x}_0^{\mathrm{KF}} \leftarrow 0$, $P_0 \leftarrow \Gamma$ (KF initialization)
  - ⋆ $F^b \leftarrow \Gamma F^\top \Gamma^{-1}$ (backward matrix)

Pseudocode: computation of $\mathcal{X}_t^{\mathrm{SCF}}$

For $t = 1, 2, \ldots$

- Read new measurement y_t
- Update KF:
  - ⋆ $K_t \leftarrow (FP_{t-1}F^\top + V)H^\top (W + H(FP_{t-1}F^\top + V)H^\top)^{-1}$
  - ⋆ $\hat{x}_t^{\mathrm{KF}} \leftarrow F\hat{x}_{t-1}^{\mathrm{KF}} + K_t(y_t - HF\hat{x}_{t-1}^{\mathrm{KF}})$
  - ⋆ $P_t \leftarrow FP_{t-1}F^\top + V - K_t(W + H(FP_{t-1}F^\top + V)H^\top)K_t^\top$
- Update the ratio matrix:
  - ⋆ $U_t A_t \leftarrow F(U_{t-1}A_{t-1})F^b + K_t(H - HF(U_{t-1}A_{t-1})F^b)$
- If $t \geq n$
  - Compute[13]
    - ⋆ $\hat{x}_t \leftarrow (U_t A_t)^{-1} \hat{x}_t^{\mathrm{KF}}$
    - ⋆ $\Pi_t \leftarrow (U_t A_t)^{-1} P_t$
    - ⋆ $\mathcal{X}_t^{\mathrm{SCF}} \leftarrow \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t)^\top \Pi_t^{-1} (x - \hat{x}_t) \leq \chi^2(\alpha, n) \right\}$, where $\chi^2(\alpha, n)$ is the quantile at probability α of the Chi-square distribution with n degrees of freedom
  - Output: $\mathcal{X}_t^{\mathrm{SCF}}$

Pseudocode: SCF initialization

- Set the probability level $\alpha \in (0,1)$
- Let
  - ⋆ $U_0 A_0 \leftarrow 0$
  - ⋆ $\hat{x}_0^{\mathrm{KF}} \leftarrow 0$, $P_0 \leftarrow \Gamma$ (KF initialization)
  - ⋆ $F^b \leftarrow \Gamma F^\top \Gamma^{-1}$ (backward matrix)

Pseudocode: computation of $\mathcal{X}_t^{\mathrm{SCF}}$

For $t = 1, 2, \ldots$

- Read new measurement y_t
- Update KF:
  - ⋆ $K_t \leftarrow (FP_{t-1}F^\top + V)H^\top (W + H(FP_{t-1}F^\top + V)H^\top)^{-1}$
  - ⋆ $\hat{x}_t^{\mathrm{KF}} \leftarrow F\hat{x}_{t-1}^{\mathrm{KF}} + K_t(y_t - HF\hat{x}_{t-1}^{\mathrm{KF}})$
  - ⋆ $P_t \leftarrow FP_{t-1}F^\top + V - K_t(W + H(FP_{t-1}F^\top + V)H^\top)K_t^\top$
- Update the ratio matrix:
  - ⋆ $U_t A_t \leftarrow F(U_{t-1}A_{t-1})F^b + K_t(H - HF(U_{t-1}A_{t-1})F^b)$
- If $t \geq n$
  - Compute[13]
    - ⋆ $\hat{x}_t \leftarrow (U_t A_t)^{-1} \hat{x}_t^{\mathrm{KF}}$
    - ⋆ $\Pi_t \leftarrow (U_t A_t)^{-1} P_t$
    - ⋆ $\mathcal{X}_t^{\mathrm{SCF}} \leftarrow \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t)^\top \Pi_t^{-1} (x - \hat{x}_t) \leq \chi^2(\alpha, n) \right\}$, where $\chi^2(\alpha, n)$ is the quantile at probability α of the Chi-square distribution with n degrees of freedom
  - Output: $\mathcal{X}_t^{\mathrm{SCF}}$

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

- DATA-INDEPENDENT MATRIX
- CAN BE COMPUTED RECURSIVELY

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

KALMAN-RELATED QUANTITIES

$(U_t A_t)^{-1}$ depends on data richness

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \left\{ x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n) \right\},$$

- DATA-INDEPENDENT MATRIX
- CAN BE COMPUTED RECURSIVELY

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

KALMAN-RELATED QUANTITIES

$(U_t A_t)^{-1}$ depends on data richness

MAIN INGREDIENTS FOR THE DERIVATION:

- Hilbert Projection Theorem + constraints

DYNAMIC CASE

$$\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$$

STATE CONDITIONAL ELLIPSOID

$$\mathcal{X}_t^{\mathrm{SCF}} = \{x \in \mathbb{R}^n : (x - \hat{x}_t^{SCF})^T \Pi_t^{-1} (x - \hat{x}_t^{SCF}) \leq \chi^2(\alpha, n)\},$$

- DATA-INDEPENDENT MATRIX
- CAN BE COMPUTED RECURSIVELY

$$\hat{x}_t^{SCF} = (U_t A_t)^{-1} \hat{x}_t$$

$$\Pi_t = (U_t A_t)^{-1} P_t$$

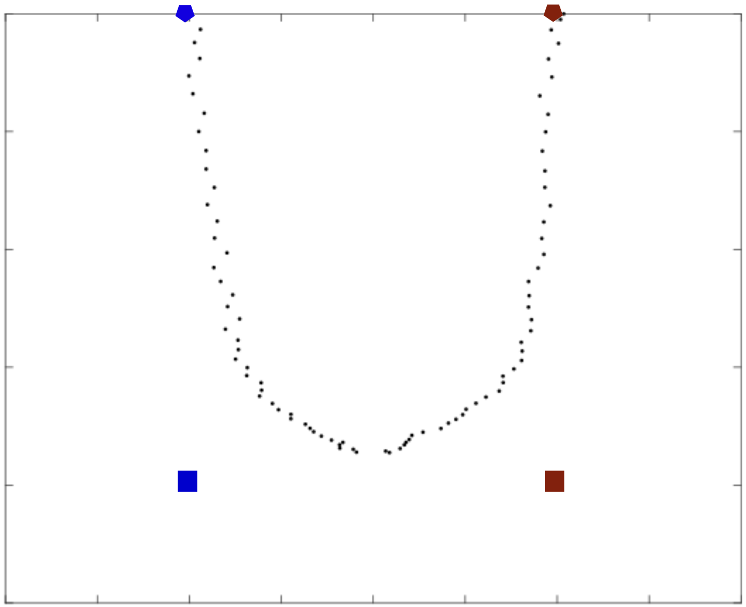
KALMAN-RELATED QUANTITIES

$(U_t A_t)^{-1}$ depends on data richness

MAIN INGREDIENTS FOR THE DERIVATION:

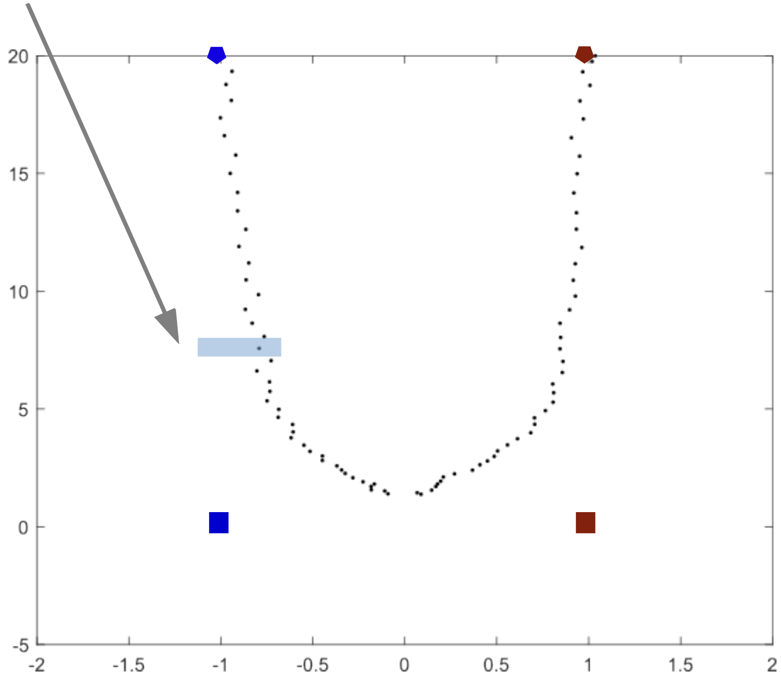
- Hilbert Projection Theorem + constraints
- Reformulating $\begin{cases} x_{t+1} = Fx_t + v_t \\ y_t = Hx_t + w_t \end{cases}$ as Backward Markov Process

CRITICAL CASE
Two robots run their **return-to-base** program



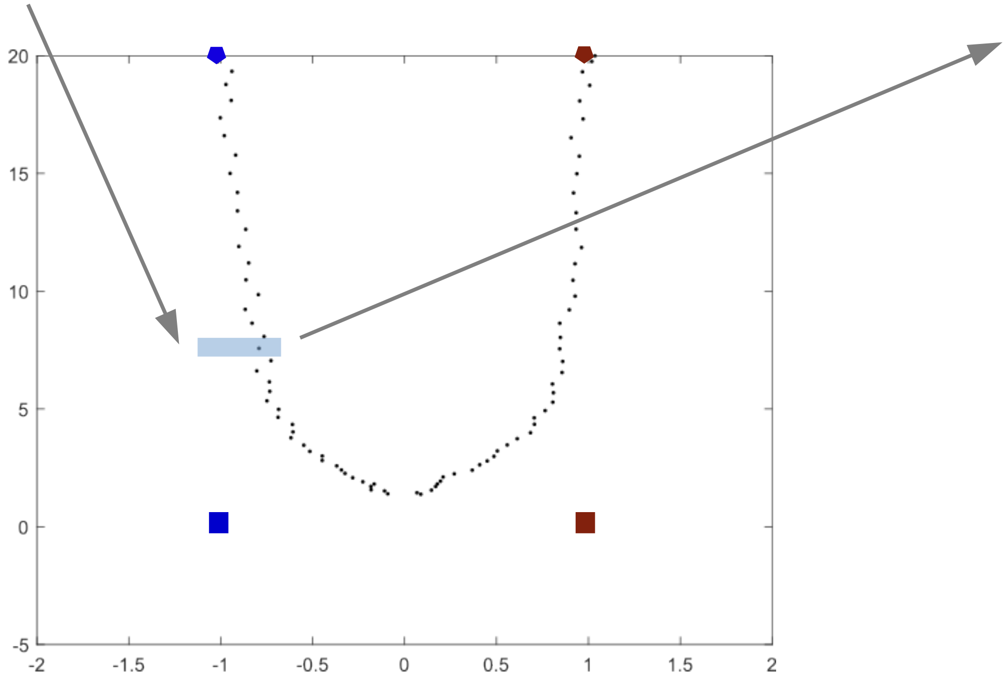
CRITICAL CASE
Two robots run their **return-to-base** program

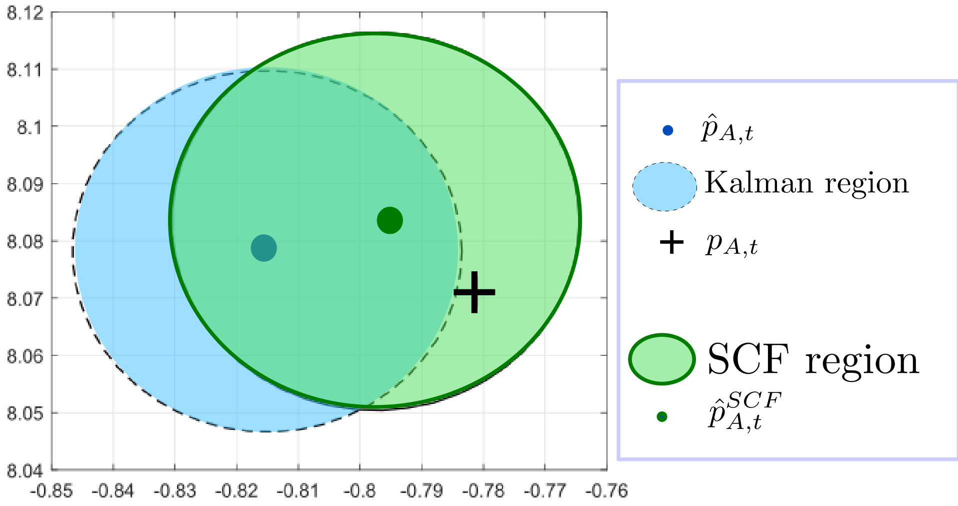
LET US ZOOM IN



CRITICAL CASE
Two robots run their **return-to-base** program

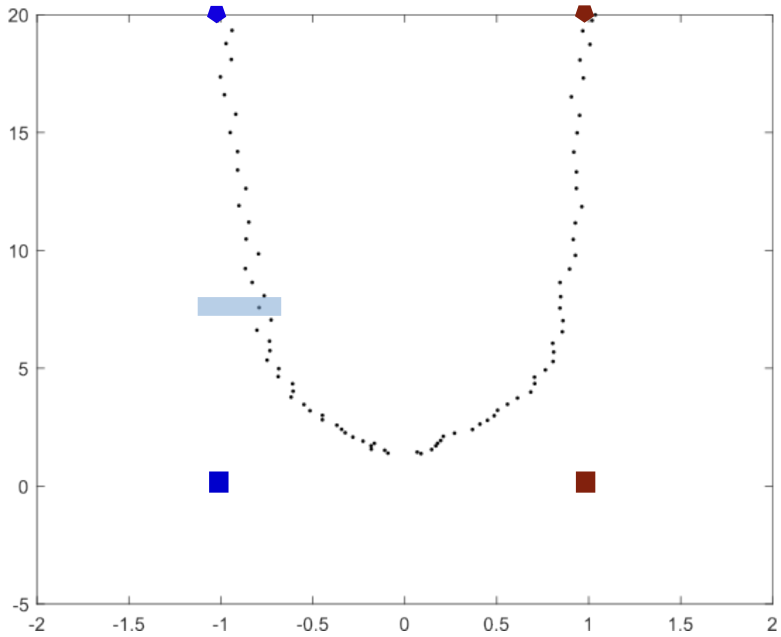
LET US ZOOM IN

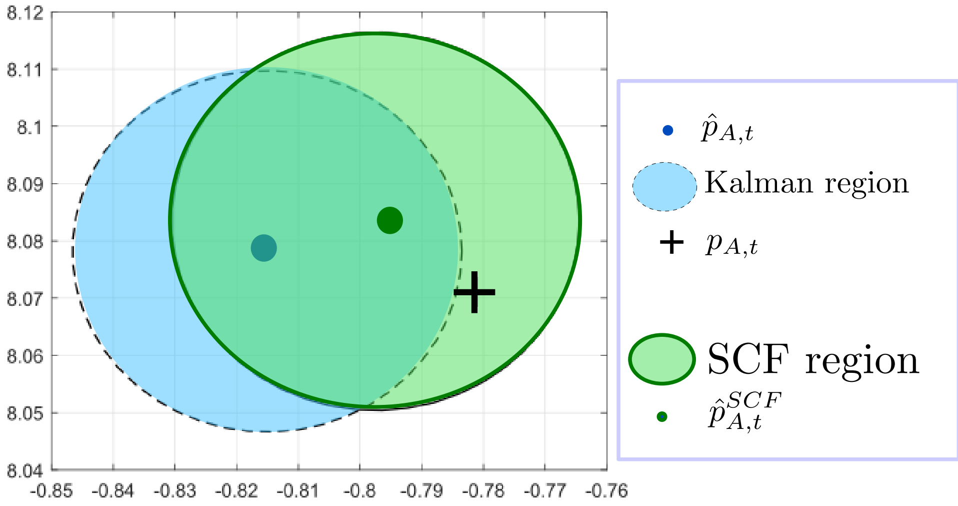




CRITICAL CASE
Two robots run their **return-to-base** program

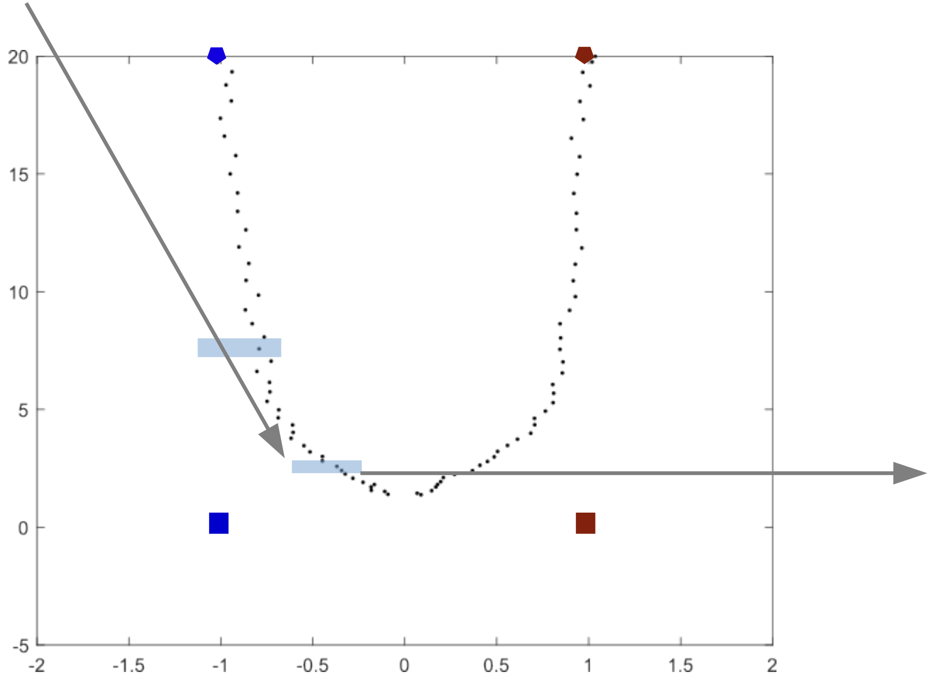
LET US ZOOM IN

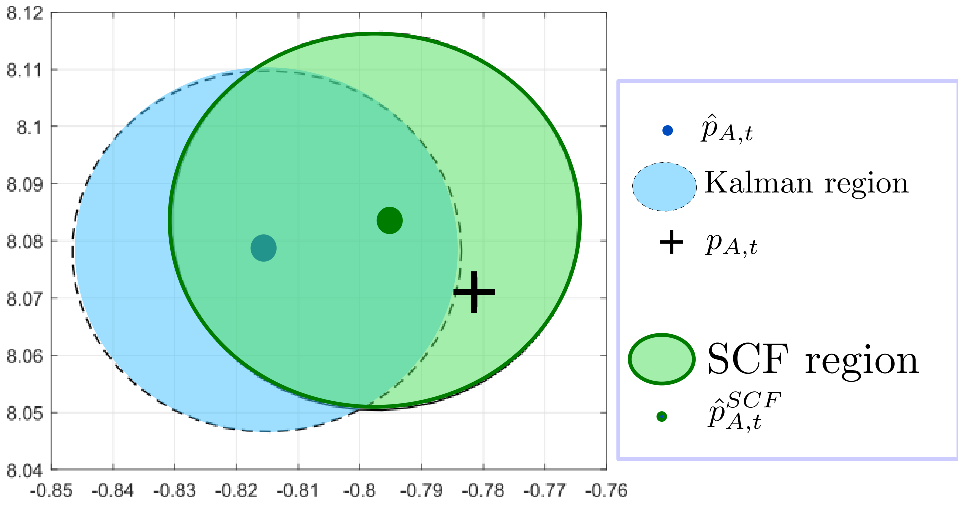


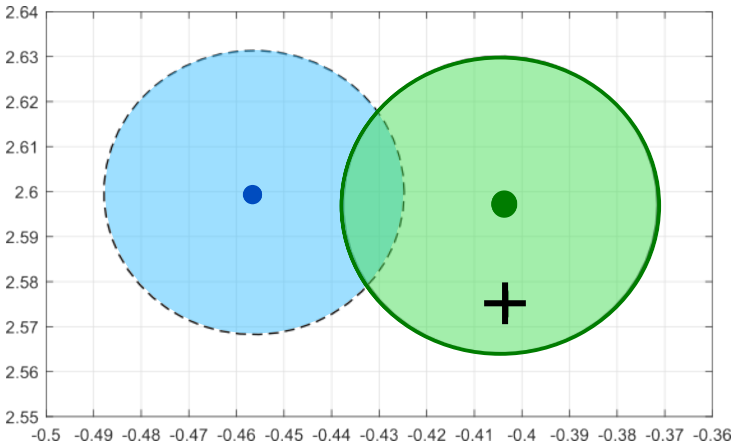


CRITICAL CASE
Two robots run their **return-to-base** program

LET US ZOOM IN







TAKE-HOME MESSAGE

TAKE-HOME MESSAGE

- In many filtering (or prediction or smoothing problems) **some values of the states are more important** than others.

TAKE-HOME MESSAGE

- In many filtering (or prediction or smoothing problems) **some values of the states are more important** than others.
- In these cases, a desirable property is that the true state is included in the constructed regions with the desired probability, regardless of its value (**STATE CONDITIONAL PROPERTY**).

TAKE-HOME MESSAGE

- In many filtering (or prediction or smoothing problems) **some values of the states are more important** than others.

- In these cases, a desirable property is that the true state is included in the constructed regions with the desired probability, regardless of its value (**STATE CONDITIONAL PROPERTY**).

- Kalman regions do not satisfy the STATE CONDITIONAL PROPERTY.

TAKE-HOME MESSAGE

- In many filtering (or prediction or smoothing problems)
  some values of the states are more important than others.

- In these cases, a desirable property is that the true state is included in the constructed regions with the desired probability, regardless of its value (**STATE CONDITIONAL PROPERTY**).

- Kalman regions do not satisfy the STATE CONDITIONAL PROPERTY.

- The STATE CONDITIONAL PROPERTY is secured by a new technique, here called the “State Conditional Filter”.

TAKE-HOME MESSAGE

- In many filtering (or prediction or smoothing problems) **some values of the states are more important** than others.

- In these cases, a desirable property is that the true state is included in the constructed regions with the desired probability, regardless of its value (**STATE CONDITIONAL PROPERTY**).

- Kalman regions do not satisfy the STATE CONDITIONAL PROPERTY.

- The STATE CONDITIONAL PROPERTY is secured by a new technique, here called the "State Conditional Filter".

There are several open problems and pending questions...

Open problems and pending questions:

OPTIMALITY

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

GENERALITY AND ROBUSTNESS

- The paradigm is general, but the **working assumptions** are limiting.

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

GENERALITY AND ROBUSTNESS

- The paradigm is general, but the **working assumptions** are limiting.
- There are some preliminary (unpublished) studies results regarding robustness against misspecifications.

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

GENERALITY AND ROBUSTNESS

- The paradigm is general, but the **working assumptions** are limiting.
- There are some preliminary (unpublished) studies results regarding robustness against misspecifications.

APPLICATION-DRIVEN TAILORING

- Sometimes you know *a priori* which states are important/unimportant/impossible.

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

GENERALITY AND ROBUSTNESS

- The paradigm is general, but the **working assumptions** are limiting.
- There are some preliminary (unpublished) studies results regarding robustness against misspecifications.

APPLICATION-DRIVEN TAILORING

- Sometimes you know *a priori* which states are important/unimportant/impossible.

OTHER DIRECTIONS

- Distributed setup, adversarial setup, ...

THANK YOU

Open problems and pending questions:

OPTIMALITY

- We know that there is no algorithm with the STATE CONDITIONAL PROPERTY that delivers regions that are **always strictly** smaller than those that we deliver. (Proof in the paper)

GENERALITY AND ROBUSTNESS

- The paradigm is general, but the **working assumptions** are limiting.
- There are some preliminary (unpublished) studies results regarding robustness against misspecifications.

APPLICATION-DRIVEN TAILORING

- Sometimes you know *a priori* which states are important/unimportant/impossible.

OTHER DIRECTIONS

- Distributed setup, adversarial setup, ...

ALGO.CARE@UNIBS.IT