Outline

1. Introduction
2. The Bayes Filter
3. Gaussian filters
4. The Kalman filter

Labs:

SLAM Toolbox with Matlab

Assessment:

Labs + Exam (100%)

4.1 The Kalman Filter

Algorithm Kalman Filter $\left(\hat{x}_{k-1}, P_{k-1}, u_k, z_k\right)$

$$\hat{x}_k^- = A_k \hat{x}_{k-1} + B_k u_k$$
$$P_k^- = A_k P_{k-1} A_k^T + Q_k$$
$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H_k \hat{x}_k^-\right)$$
$$P_k = \left(I - K_k H_k\right) P_k^-$$
$$return\left(\hat{x}_k, P_k\right)$$

Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$$

Observation:

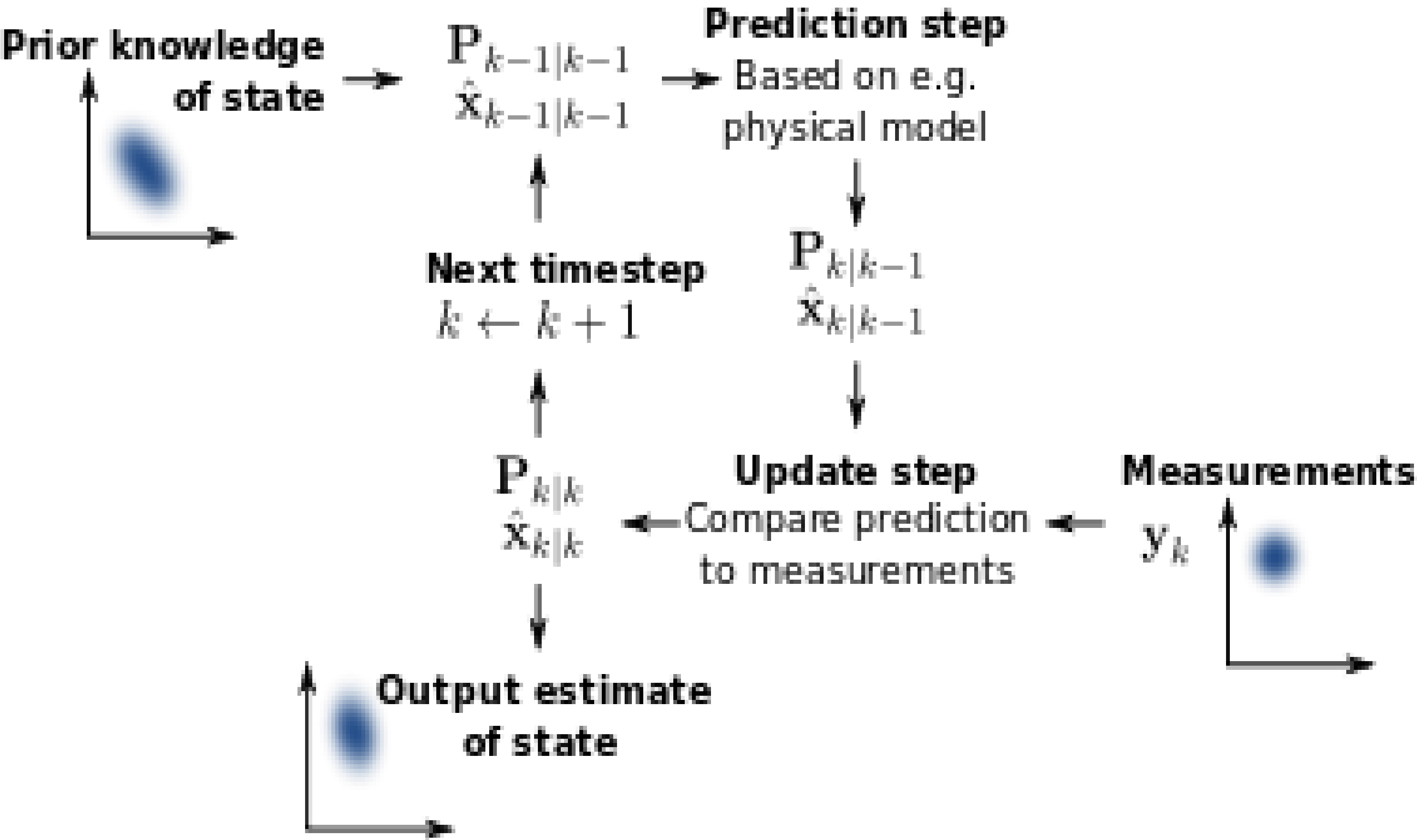
$$\mathbf{z}_k$$

Update:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$
$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$
$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

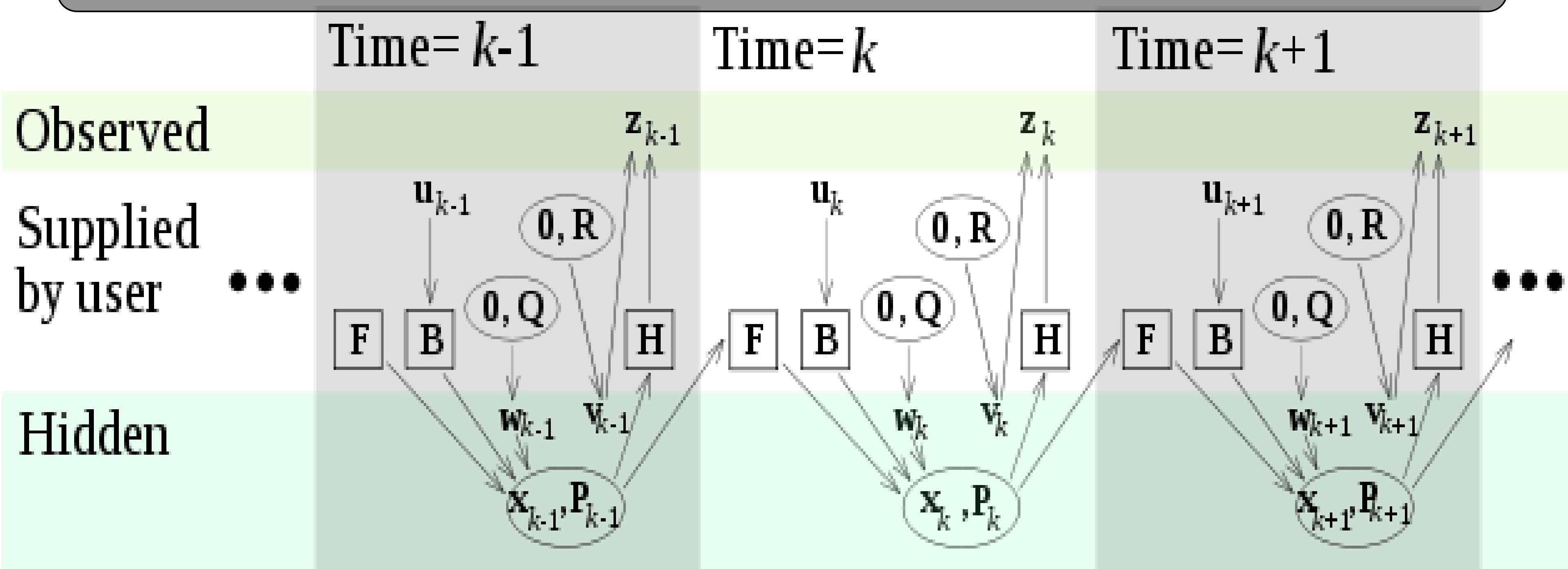
From Wikipedia

4.1 The Kalman Filter



From Wikipedia

4.1 The Kalman Filter



vectors

- ***x**$_k$: state*

- ***u**$_k$: control input*
- ***z**$_k$: observation*
- ***w**$_k$: process noise = N(0,**Q**$_k$)*
- ***v**$_k$: observation noise = N(0,**R**$_k$)*

matrices

- ***P**$_k$: covariance of the state*
- ***F**$_k$: state-transition model*
- ***B**$_k$: control-input model*
- ***H**$_k$: observation model*
- ***Q**$_k$: covariance of the process noise*
- ***R**$_k$: covariance of the observation noise*

From Wikipedia

A step forward → SLAM

Algorithm: SLAM

$\{$- Pose initialization$\}$

$x_0^B = \hat{x}_0^B;\ \mathrm{P}_0^B = \hat{\mathrm{P}}_0^B;$

$\left[z_0, R_0\right] = get\_measurements;$

$\left[x_0^B, P_0^B\right] = add\_new\_features\left(x_0^B, P_0^B, z_0, R_0\right);$

for k=1 to steps **do**

$\left[u_{R_k}^{R_{k-1}}, Q_k\right] = get\_odometry$

$\{$-EKF prediction$\}$

$\left[x_{k|k-1}^B, P_{k|k-1}^B\right] = move\_vehicle\left(x_{k-1}^B, P_{k-1}^B, u_{R_k}^{R_{k-1}}, Q_k\right)$

$\left[z_k, R_k\right] = get\_measurements$

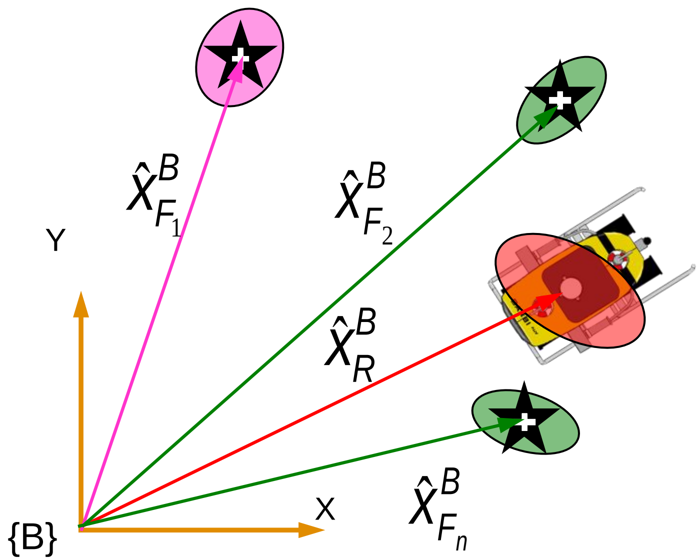
$\mathcal{H}_k = data\_association\left(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k\right);$

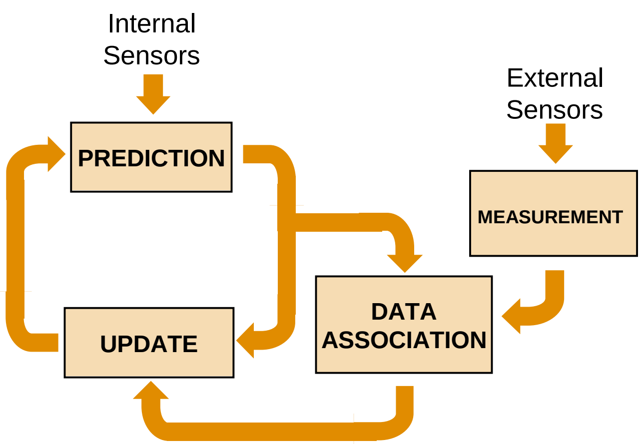
$\{$-EKF update$\}$

$\left[x_k^B, P_k^B\right] = update\_position\left(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k\right);$

$\left[x_k^B, P_k^B\right] = add\_new\_features\left(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k\right);$

end for

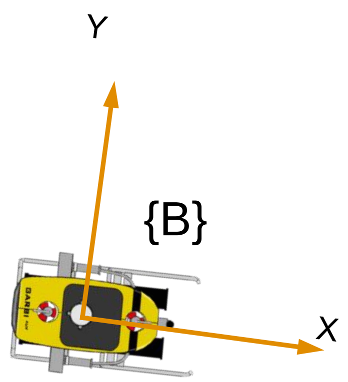


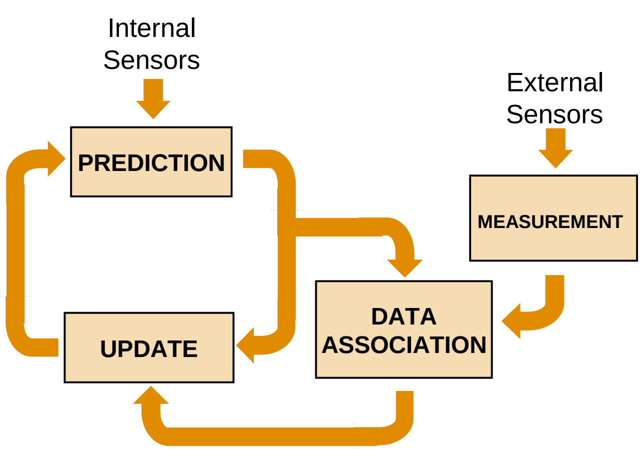


Spatial Relationships

$$\hat{x}^B = E\left[x^B\right] = \begin{bmatrix} \hat{x}^B_R \\ \hat{x}^B_{F_1} \\ \hat{x}^B_{F_2} \\ \vdots \\ \hat{x}^B_{F_n} \end{bmatrix} \qquad P^B = E\left[\left(x^B - \hat{x}^B\right)\left(x^B - \hat{x}^B\right)^T\right] = \begin{bmatrix} P^B_R & P^B_{RF_1} & P^B_{RF_2} & \cdots & P^B_{RF_n} \\ P^B_{F_1R} & P^B_{F_1} & P^B_{F_1F_2} & \cdots & P^B_{F_1F_n} \\ P^B_{F_2R} & P^B_{F_2F_1} & P^B_{F_2} & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P^B_{F_nR} & P^B_{F_nF_1} & & \cdots & P^B_{F_n} \end{bmatrix}$$

- A map is represented by:
  - The robot pose referenced to {B}
  - The location of a set of features referenced to {B}
  - The covariances between them
- The map is built incrementally





SLAM Algorithm

Algorithm: SLAM

{- Pose initialization}

$x_0^B = \hat{x}_0^B;\ \mathrm{P}_0^B = \hat{\mathrm{P}}_0^B;$

$\left[z_0, R_0\right] = get\_measurements;$

$\left[x_0^B, P_0^B\right] = add\_new\_features\left(x_0^B, P_0^B, z_0, R_0\right);$

for k=1 to steps **do**

$\left[u_{R_k}^{R_{k-1}}, Q_k\right] = get\_odometry$

{-EKF prediction}

$\left[x_{k|k-1}^B, P_{k|k-1}^B\right] = move\_vehicle\left(x_{k-1}^B, P_{k-1}^B, x_{R_k}^{R_{k-1}}, Q_k\right)$

$\left[z_k, R_k\right] = get\_measurements$

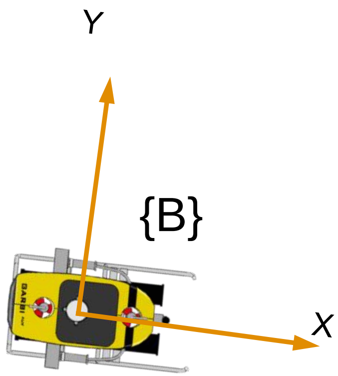
$\mathcal{H}_k = data\_association\left(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k\right);$

{-EKF update}

$\left[x_k^B, P_k^B\right] = update\_position\left(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k\right);$

$\left[x_k^B, P_k^B\right] = add\_new\_features\left(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k\right);$

end for

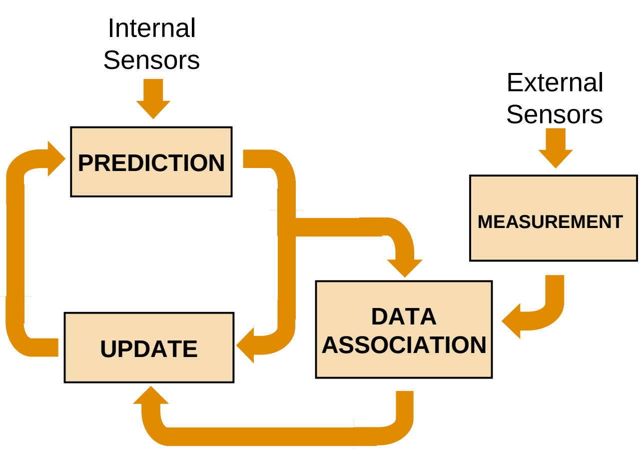


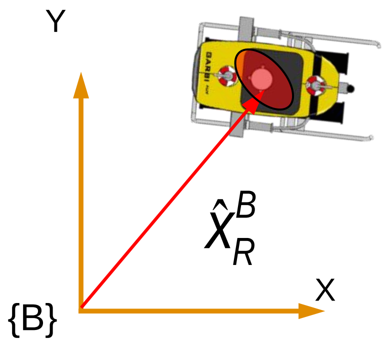
Initialization (II)

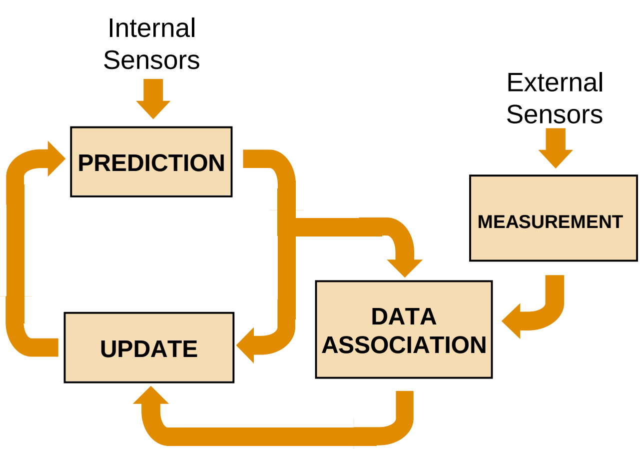
$$\hat{x}^B = \begin{bmatrix} 0 \\ \\ \\ \\ \end{bmatrix}, \quad P^B = \begin{bmatrix} 0 & & \\ & & \\ & & \\ & & \end{bmatrix}$$

> The process starts Selecting the current position of the robot as the Reference frame

$$\hat{x}_R^B = 0 \qquad P_R^B = 0$$







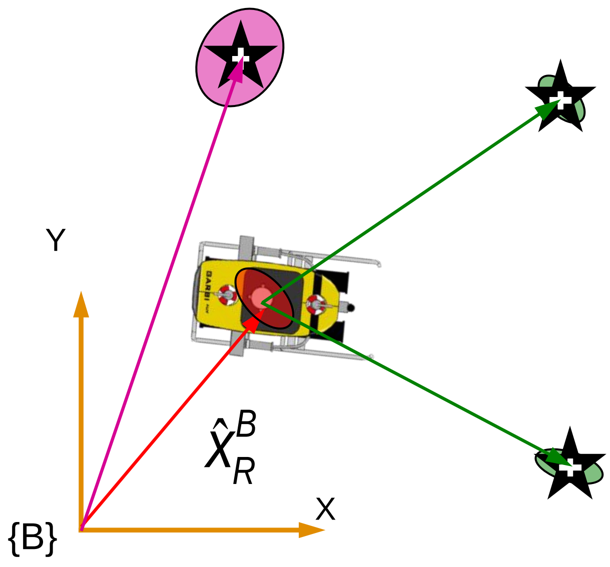
Initialization (III)

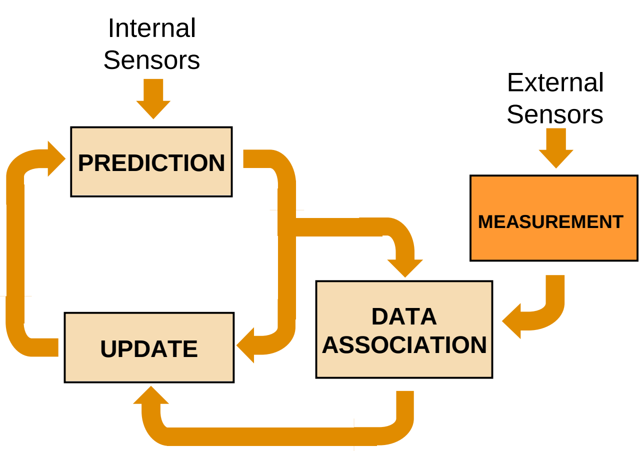
$$\hat{X}_k^B = \begin{bmatrix} \hat{X}_R^B \\ \\ \\ \\ \end{bmatrix} , \quad P_k^B = \begin{bmatrix} P_R^B & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

> Sometimes it is interesting to select another frame as the global reference

> In this case an estimation of the position mean and covariance is needed: $\hat{X}_R^B \quad P_R^B$

Adding New Features





Algorithm: SLAM

{- Pose initialization}

$x_0^B = \hat{x}_0^B$; $P_0^B = \hat{P}_0^B$;

$[z_0, R_0] = get\_measurements;$

$[x_0^B, P_0^B] = add\_new\_features(x_0^B, P_0^B, z_0, R_0);$

for k=1 to steps **do**

$[u_{R_k}^{R_{k-1}}, Q_k] = get\_odometry$

{-EKF prediction}

$[x_{k|k-1}^B, P_{k|k-1}^B] = move\_vehicle(x_{k-1}^B, P_{k-1}^B, x_{R_k}^{R_{k-1}}, Q_k)$

$[z_k, R_k] = get\_measurements$

$\mathcal{H}_k = data\_association(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

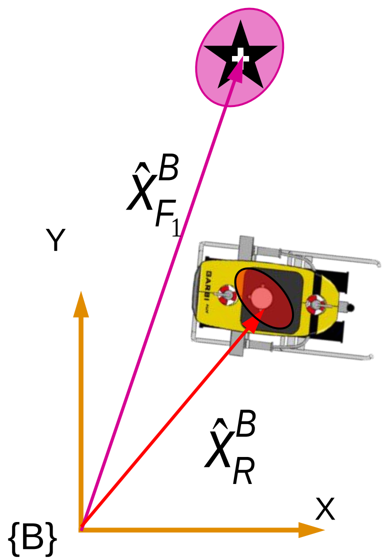
{-EKF update}

$[x_k^B, P_k^B] = update\_position(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

$[x_k^B, P_k^B] = add\_new\_features(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k);$

end for

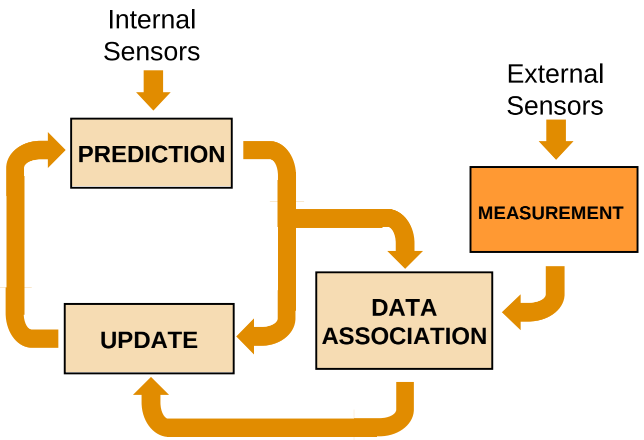
Adding New Features: World Referenced



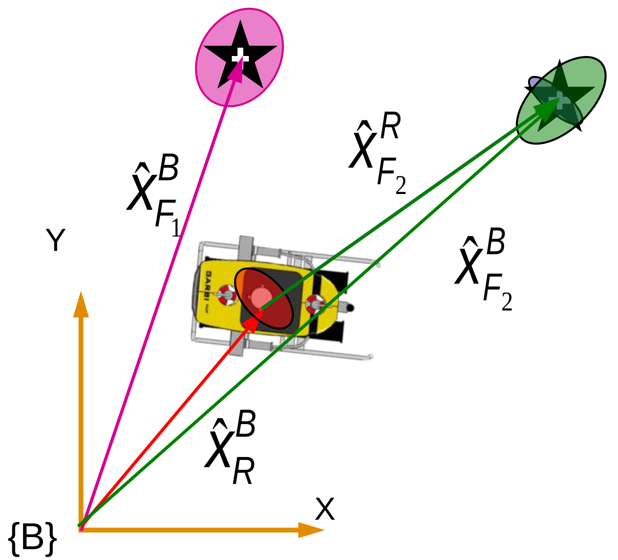
$$\hat{X}_k^B = \begin{bmatrix} \hat{X}_R^B \\ \hat{X}_{F_1}^B \\ \\ \\ \end{bmatrix}, \quad P_k^B = \begin{bmatrix} P_R^B & 0 & \\ 0 & P_{F_1}^B & \\ & & \\ \end{bmatrix}$$

> A New Feature is observed: $\hat{X}_{F_1}^B \quad P_{F_1}^B$

> Feature & Robot position are uncorrelated



Adding New Features: Robot Referenced



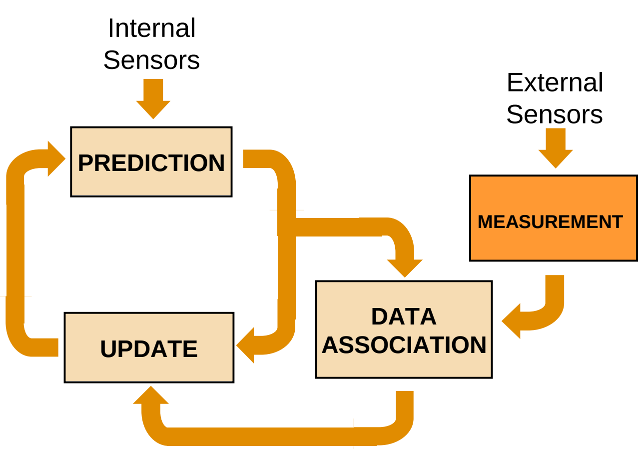
$$\hat{X}_k^B = \begin{bmatrix} \hat{X}_R^B \\ \hat{X}_{F_1}^B \\ \hat{X}_R^B \oplus \hat{X}_{F_2}^R \end{bmatrix}, \quad P_k^B = \begin{bmatrix} P_R^B & 0 & ? \\ 0 & P_{F_1}^B & ? \\ ? & ? & P_{F_2}^B \end{bmatrix}$$

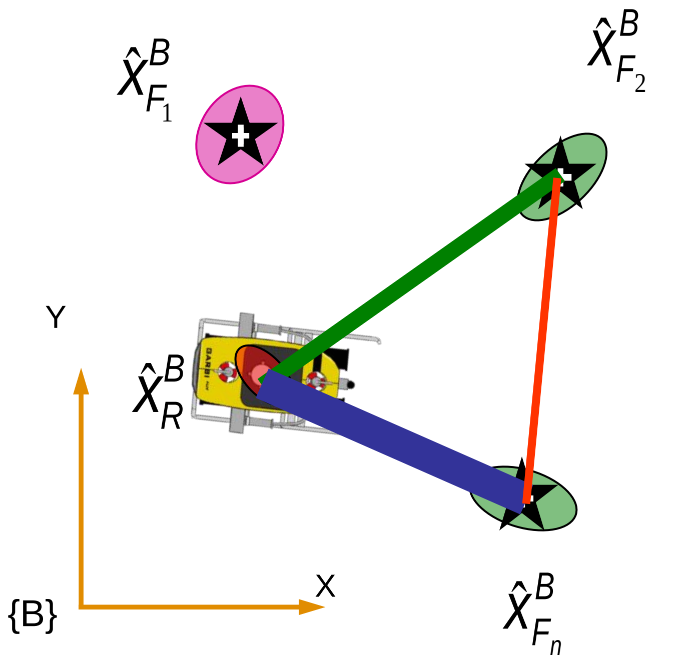
> A New Feature is observed: $\hat{X}_{F_2}^R$ R_k

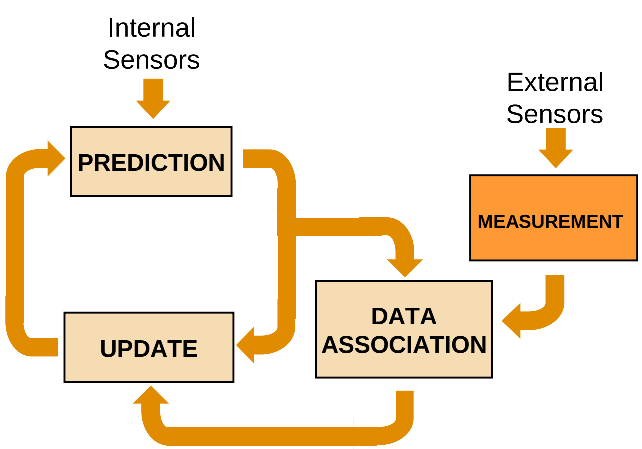
> Feature uncertainty includes the noise of the robot & the noise of the measurement.

$$\hat{X}_{F_2}^B = \hat{X}_R^B \oplus \hat{X}_{F_2}^R$$

$$P_{F_2}^B = J_{1\oplus} P_R^B J_{1\oplus}^T + J_{2\oplus} R_k J_{2\oplus}^T$$







What does correlation mean?

> Imagine correlation as a network of springs interconnecting all the features

$$P^B = \begin{bmatrix} P^B_R & P^B_{RF_1} & P^B_{RF_2} & \cdots & P^B_{RF_n} \\ P^B_{F_1R} & P^B_{F_1} & P^B_{F_1F_2} & \cdots & P^B_{F_1F_n} \\ P^B_{F_2R} & P^B_{F_2F_1} & P^B_{F_2} & \cdots & P^B_{F_2F_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P^B_{F_nR} & P^B_{F_nF_1} & P^B_{F_nF_2} & \cdots & P^B_{F_n} \end{bmatrix}$$

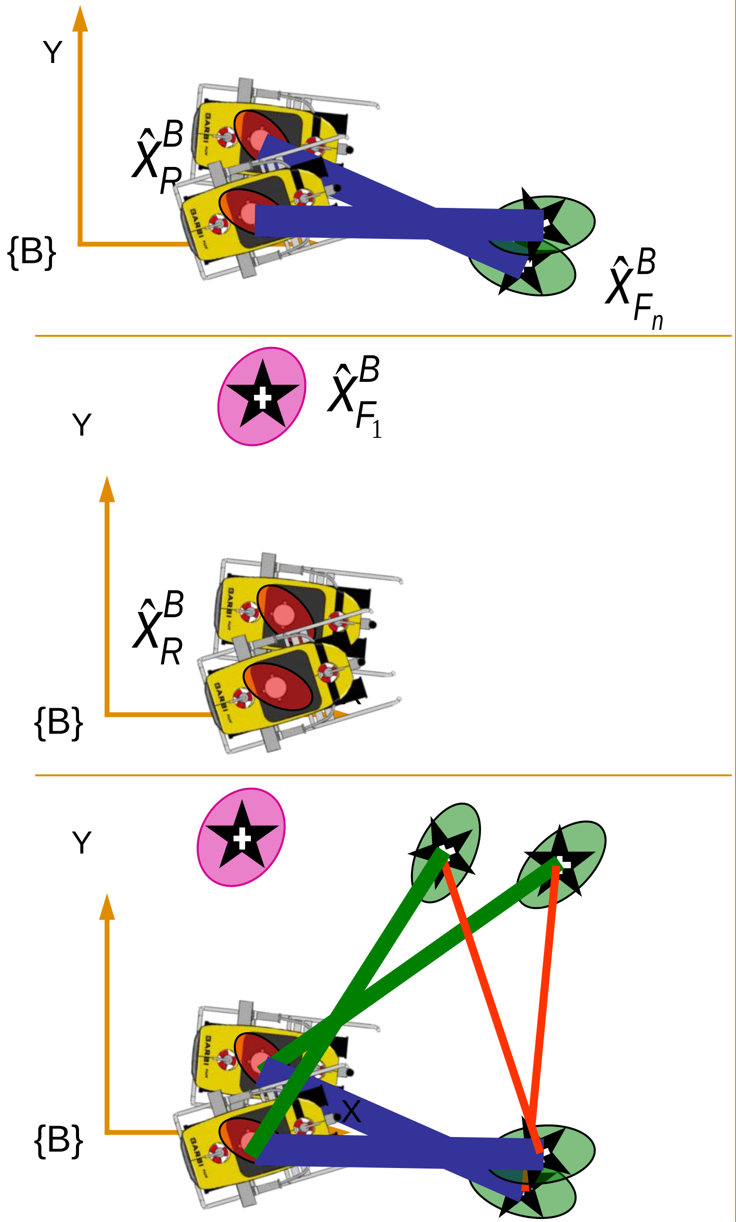
> Do you remember the correlation matrix?

$$r = \begin{bmatrix} 1 & r_{x_Ry_R} & r_{x_Rq_R} & r_{x_Rx_F} & r_{x_Ry_F} \\ r_{y_Rx_R} & 1 & r_{y_Rq_R} & r_{y_Rx_F} & r_{y_Ry_F} \\ r_{q_Rx_R} & r_{q_Ry_R} & 1 & r_{q_Rx_F} & r_{q_Ry_F} \\ r_{x_Fx_R} & r_{x_Fy_R} & r_{x_Fq_R} & 1 & r_{x_Fy_F} \\ r_{y_Fx_R} & r_{y_Fy_R} & r_{y_Fq_R} & r_{y_Fx_F} & 1 \end{bmatrix} \qquad \rho_{12} \triangleq \frac{\sigma^2_{x_1x_2}}{\sigma_{x_1}\sigma_{x_2}}$$

> $\uparrow$ Covariance $\Rightarrow \uparrow$ Correlation

> $\downarrow$ Covariance $\Rightarrow \downarrow$ Correlation

> $\uparrow$Correlation $\Rightarrow \uparrow$ Stiffness of the spring



What does correlation mean?

Full Correlation

> When fully correlated $r_{RF_n} = \pm 1$ the spring belongs rigid

> If we update the Robot pose, the feature moves together

Uncorrelation

> When uncorrelated $r_{RF_n} = 0$ the spring disappears

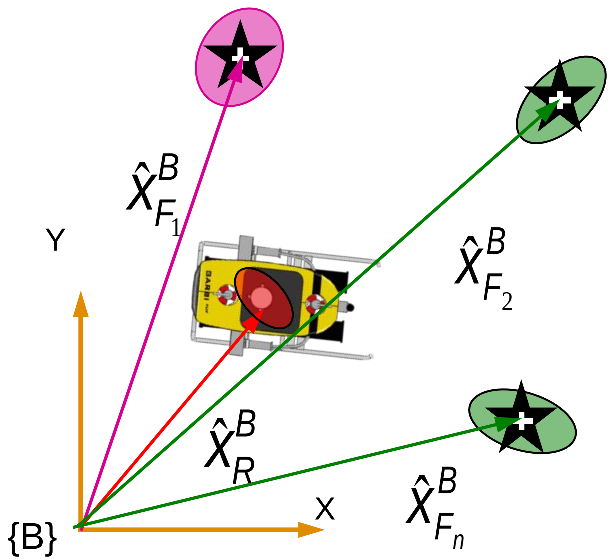
> If we update the Robot pose, the feature remains static

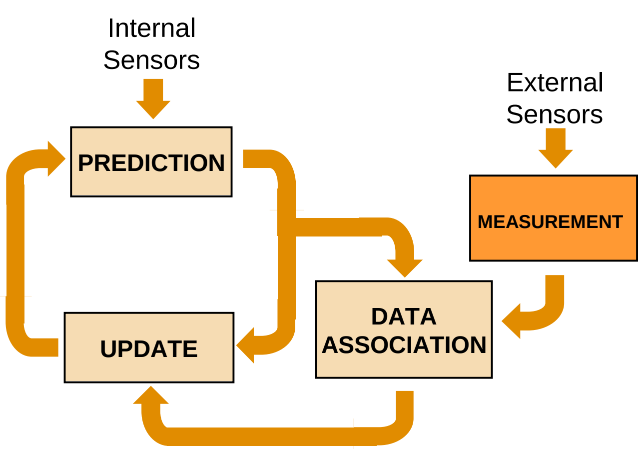
Partial Correlation

> When partially correlated $0 < \left|\rho_{RF_n}\right| < 1$:

$\uparrow$Correlation $\Rightarrow$ $\uparrow$ Stiffness of the spring

> If we update the Robot pose, the network of springs adapt the position of the features

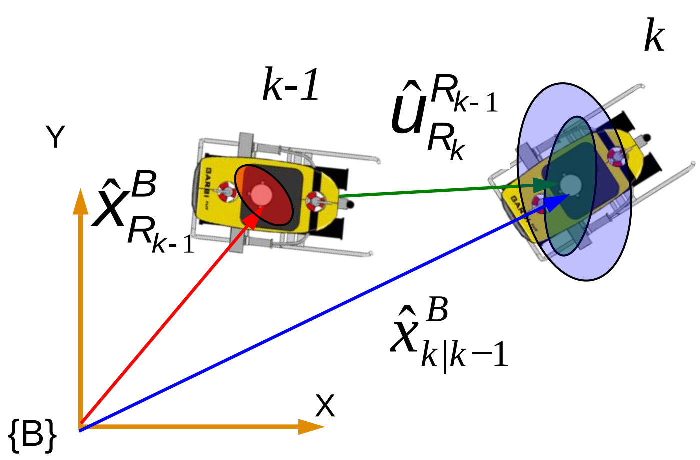


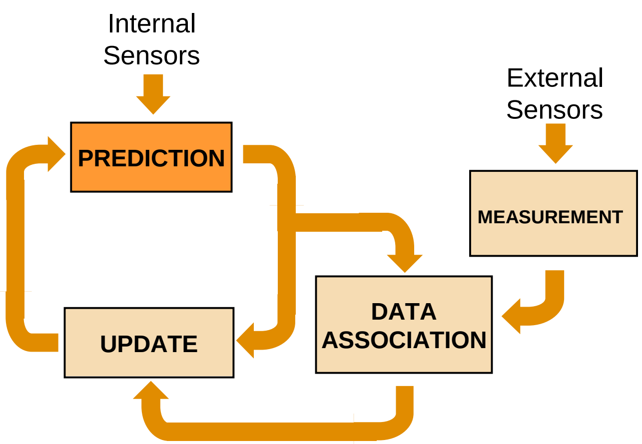


Adding New Features

$$\hat{x}^B = \begin{bmatrix} \hat{x}_R^B \\ \hat{x}_{F_1}^B \\ \hat{x}_{F_2}^B \\ \vdots \\ \hat{x}_{F_n}^B \end{bmatrix} \quad , \quad P^B = \begin{bmatrix} P_R^B & P_{RF_1}^B & P_{RF_2}^B & \cdots & P_{RF_n}^B \\ P_{F_1R}^B & P_{F_1}^B & P_{F_1F_2}^B & \cdots & P_{F_1F_n}^B \\ P_{F_2R}^B & P_{F_2F_1}^B & P_{F_2}^B & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{F_nR}^B & P_{F_nF_1}^B & & \cdots & P_{F_n}^B \end{bmatrix}$$

> This process is repeated whenever a new feature is observed





State Prediction

Algorithm: SLAM

$\{$- Pose initialization$\}$

$x_0^B = \hat{x}_0^B;\ \mathrm{P}_0^B = \hat{\mathrm{P}}_0^B;$

$[z_0, R_0] = get\_measurements;$

$[x_0^B, P_0^B] = add\_new\_features(x_0^B, P_0^B, z_0, R_0);$

for k=1 to steps **do**

$[u_{R_k}^{R_{k-1}}, Q_k] = get\_odometry$

$\{$-EKF prediction$\}$

$[x_{k|k-1}^B, P_{k|k-1}^B] = move\_vehicle(x_{k-1}^B, P_{k-1}^B, u_{R_k}^{R_{k-1}}, Q_k)$

$[z_k, R_k] = get\_measurements$

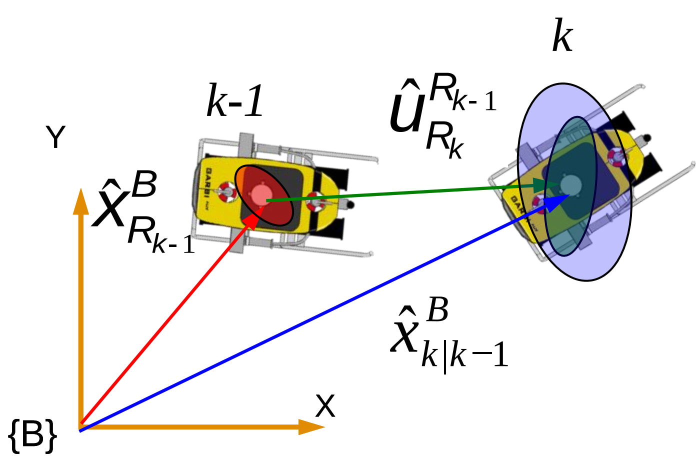
$\mathcal{H}_k = data\_association(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

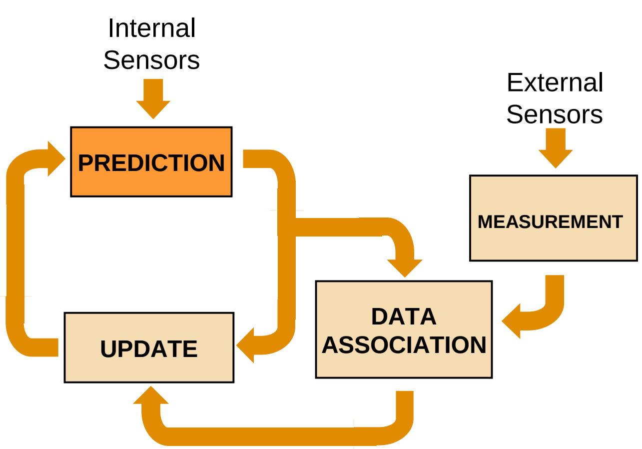
$\{$-EKF update$\}$

$[x_k^B, P_k^B] = update\_position(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

$[x_k^B, P_k^B] = add\_new\_features(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k);$

end for





State Prediction

$$\hat{x}^B_{k|k-1} = \begin{bmatrix} \hat{x}^B_{R_{k-1}} \oplus \hat{u}^{R_{k-1}}_{R_k} \\ \hat{x}^B_{R_{k-1}} \\ \vdots \\ \hat{x}^B_{F_n} \end{bmatrix} \quad P^B_{k|k-1} = \begin{bmatrix} P^B_R & ? & ? & \cdots & ? \\ ? & P^B_{F_1} & P^B_{F_1F_2} & \cdots & P^B_{F_1F_n} \\ ? & P^B_{F_2F_1} & P^B_{F_2} & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ? & P^B_{F_nF_1} & & \cdots & P^B_{F_n} \end{bmatrix}$$

> Next Pose is the previous one + the desplacement

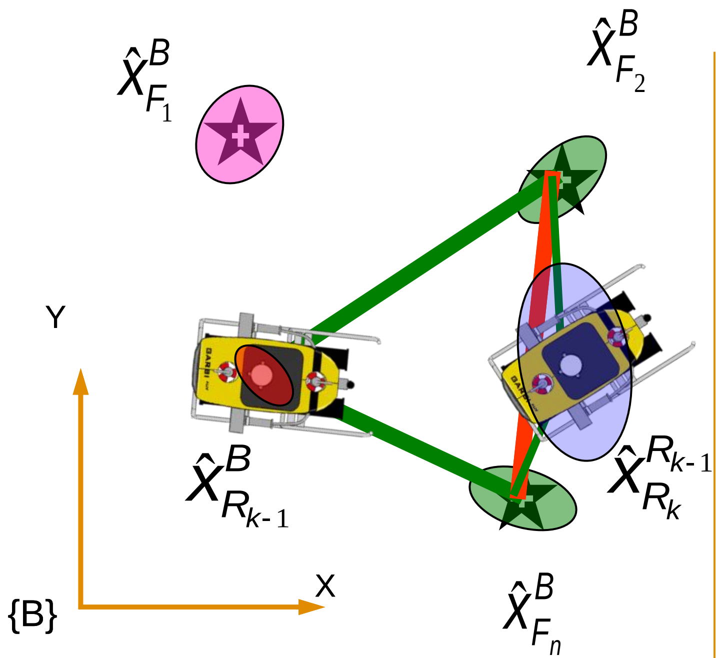
$$\hat{x}^B_{k|k-1} = \hat{x}^B_{R_{k-1}} \oplus \hat{u}^{R_{k-1}}_{R_k}$$

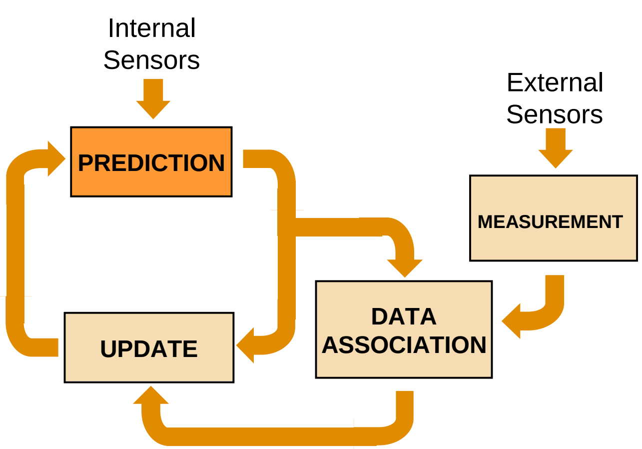
> Covariance inflation

$$P^B_{k|k-1} = F_k P^B_{k-1} F_k^T + G_k Q_k G_k^T$$

> Features are static

> $\boldsymbol{F_k}$ & $\boldsymbol{G_k}$ are the model Jacobians





State Prediction

What happens to the correlations?

> The correlations of the features with respect to the robot decrease

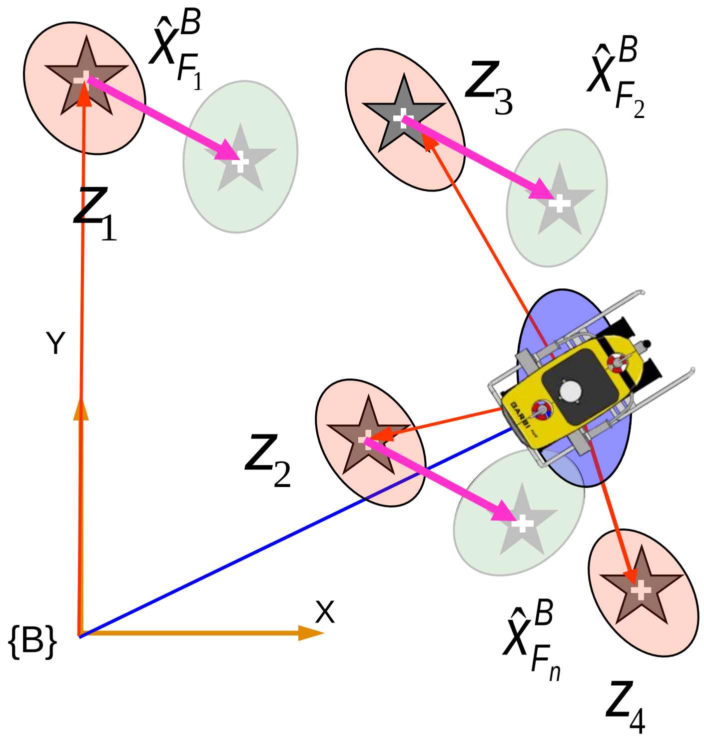
$$F_k = \begin{bmatrix} J_{1\oplus}\left\{\hat{x}^B_{R_{k-1}}, \hat{u}^{R_{k-1}}_{R_k}\right\} & 0 & \cdots & 0 \\ 0 & I & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix} \qquad G_k = \begin{bmatrix} J_{2\oplus}\left\{\hat{x}^B_{R_{k-1}}, \hat{u}^{R_{k-1}}_{R_k}\right\} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

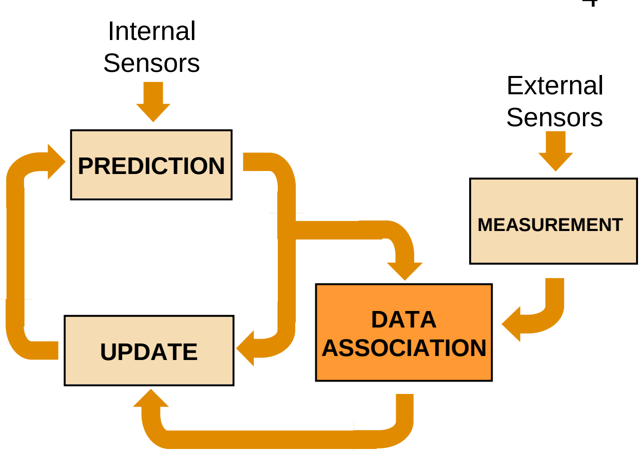
$$P^B_{k|k-1} = F_k P^B_{k-1} F_k^T + G_k Q_k G_k^T$$

$$P^B_{k|k-1} = \begin{bmatrix} J_{1\oplus}P^B_R J^T_{1\oplus} & J_{1\oplus}P^B_{RF_1} & J_{1\oplus}P^B_{RF_2} & \cdots & J_{1\oplus}P^B_{RF_n} \\ P^B_{F_1R}J^T_{1\oplus} & P^B_{F_1} & P^B_{F_1F_2} & \cdots & P^B_{F_1F_n} \\ P^B_{F_2R}J^T_{1\oplus} & P^B_{F_2F_1} & P^B_{F_2} & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P^B_{F_nR}J^T_{1\oplus} & P^B_{F_nF_1} & & \cdots & P^B_{F_n} \end{bmatrix} + \begin{bmatrix} J_{2\oplus}Q_k J^T_{2\oplus} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

> The correlations among the features does not change

4.2 SLAM





Data Association

Algorithm: SLAM

$\{$- Pose initialization$\}$

$x_0^B = \hat{x}_0^B;\ \mathrm{P}_0^B = \hat{\mathrm{P}}_0^B;$

$[z_0, R_0] = get\_measurements;$

$[x_0^B, P_0^B] = add\_new\_features(x_0^B, P_0^B, z_0, R_0);$

for k=1 to steps **do**

$[u_{R_k}^{R_{k-1}}, Q_k] = get\_odometry$

$\{$-EKF prediction$\}$

$[x_{k|k-1}^B, P_{k|k-1}^B] = move\_vehicle(x_{k-1}^B, P_{k-1}^B, x_{R_k}^{R_{k-1}}, Q_k)$

$[z_k, R_k] = get\_measurements$

$\mathcal{H}_k = data\_association(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

$\{$-EKF update$\}$

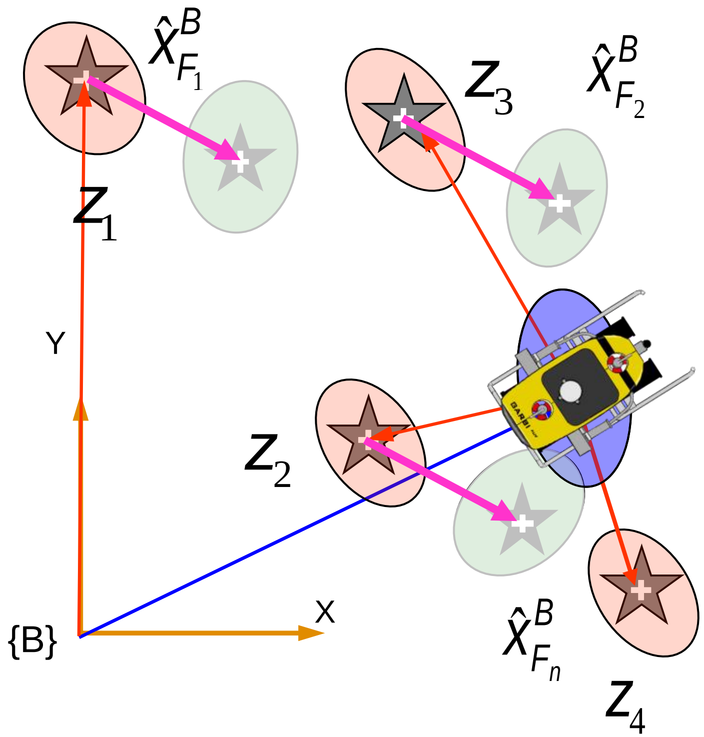
$[x_k^B, P_k^B] = update\_position(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k, \mathcal{H}_k);$

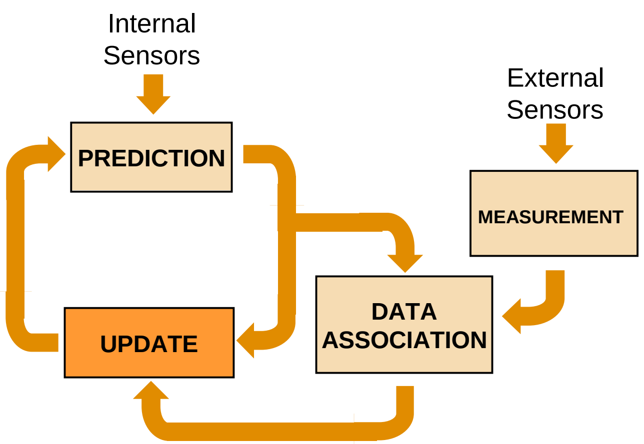
$[x_k^B, P_k^B] = add\_new\_features(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k);$

end for

$$z_k = [z_1,\ z_2,\ z_3\ ,\ z_4]$$

$$\mathcal{H}_k = [F_1,\ F_n,\ F_2,\ 0\ ]$$





State Update

Algorithm: SLAM

{- Pose initialization}

$x_0^B = \hat{x}_0^B$; $P_0^B = \hat{P}_0^B$;

$[z_0, R_0] = get\_measurements;$

$[x_0^B, P_0^B] = add\_new\_features(x_0^B, P_0^B, z_0, R_0);$

for k=1 to steps **do**

$[u_{R_k}^{R_{k-1}}, Q_k] = get\_odometry$

{-EKF prediction}

$[x_{k|k-1}^B, P_{k|k-1}^B] = move\_vehicle(x_{k-1}^B, P_{k-1}^B, x_{R_k}^{R_{k-1}}, Q_k)$

$[z_k, R_k] = get\_measurements$

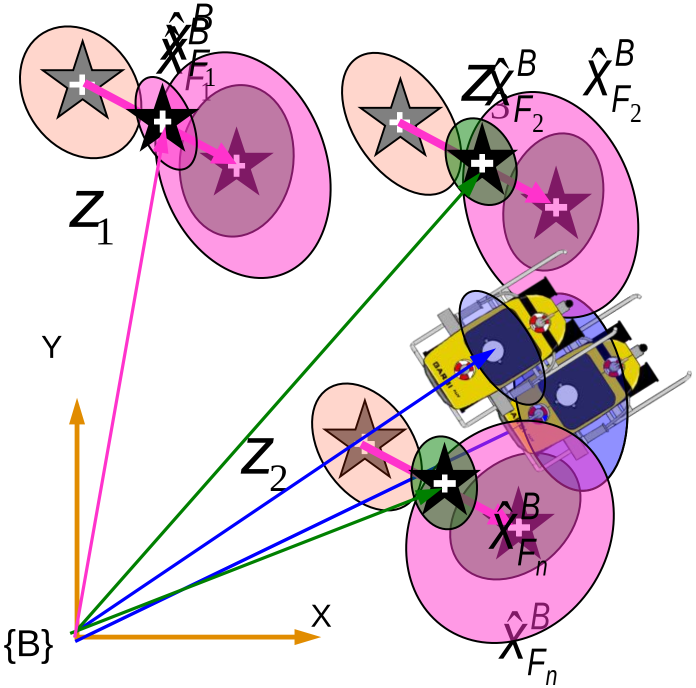
$\mathcal{H}_k = data\_association(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

{-EKF update}

$[x_k^B, P_k^B] = update\_position(x_{k|k-1}^B, P_{k|k-1}^B, z_k, R_k);$

$[x_k^B, P_k^B] = add\_new\_features(x_k^B, P_k^B, z_k, R_k, \mathcal{H}_k);$

end for



State Update

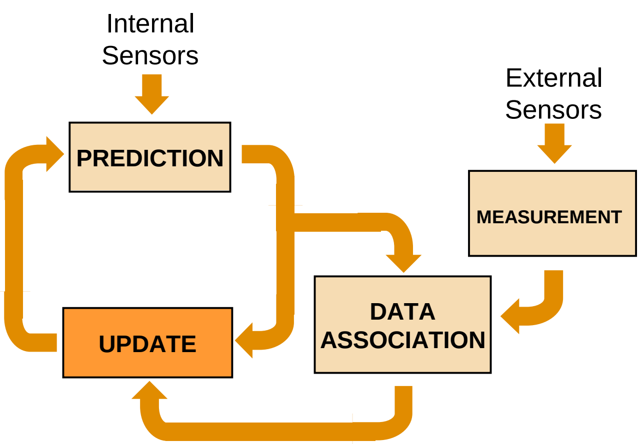
> First the discrepancies (innovation) between the features and the measurements are computed.

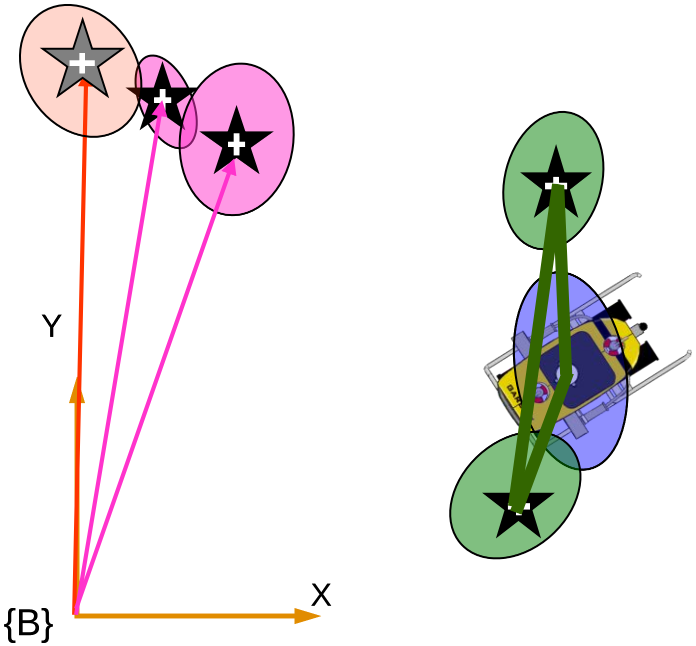
$$n_{H_k} = z_k - h_{H_k}\left(\hat{x}^B_{k|k-1}\right)$$
$$S_{H_k} = H_{H_k} P^B_{k|k-1} H^T_{H_k} + R_k$$

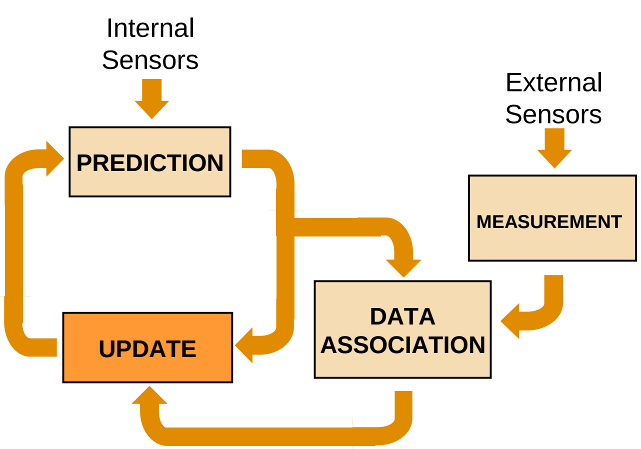
> Then the state is updated

$$K_{H_k} = P^B_{k|k-1} H^T_{H_k} S^{-1}_{H_k}$$
$$\hat{x}^B_k = \hat{x}^B_{k|k-1} + K_{H_k} n_{H_k}$$
$$P^B_k = \left(I - K_{H_k} H_{H_k}\right) P^B_{k|k-1}$$
$$= \left(I - K_{H_k} H_{H_k}\right) P^B_{k|k-1} \left(I - K_{H_k} H_{H_k}\right)^T + K_{H_k} R_k K^T_{H_k}$$

> $\boldsymbol{H_k}$ is the jacobian of the measurement function







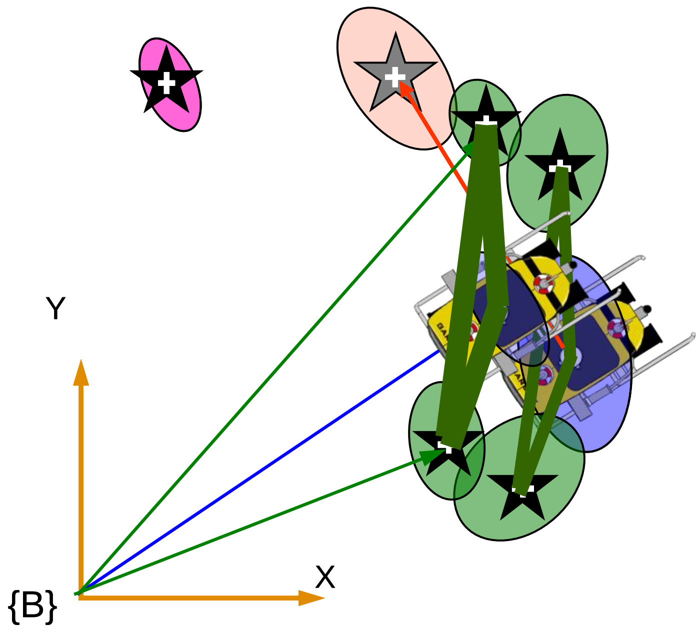
What happens to the correlations?

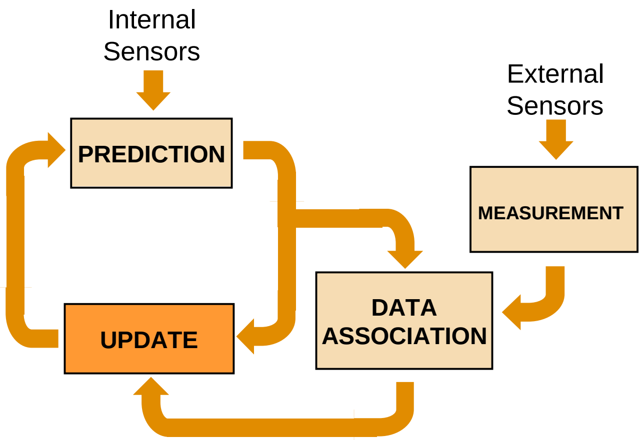
Observation of an uncorrelated feature

> The uncertainty of the feature decrease

> No other feature nor the robot decrease their uncertainty

> There are no changes in the correlations





What happens to the correlations?

Observation of an uncorrelated feature

> The uncertainty of the feature decrease

> No other feature nor the robot decrease their uncertainty

> There are no changes in the correlations

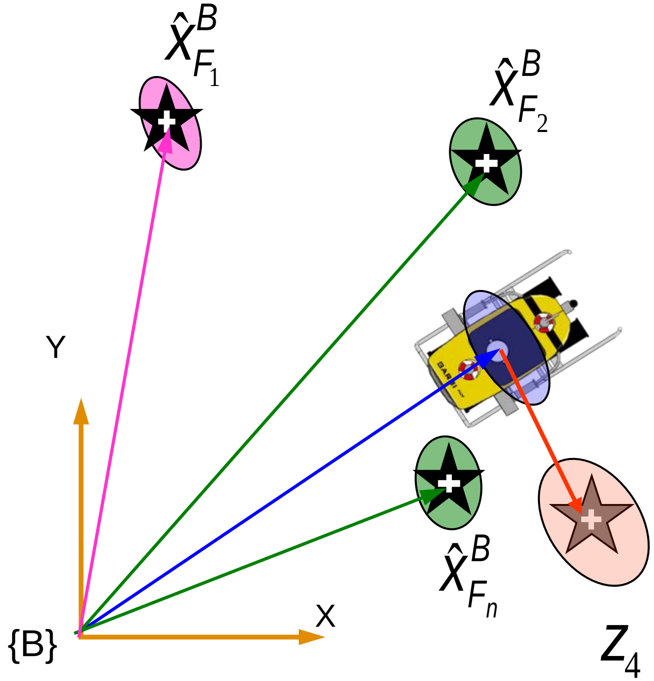
Observation of a correlated feature

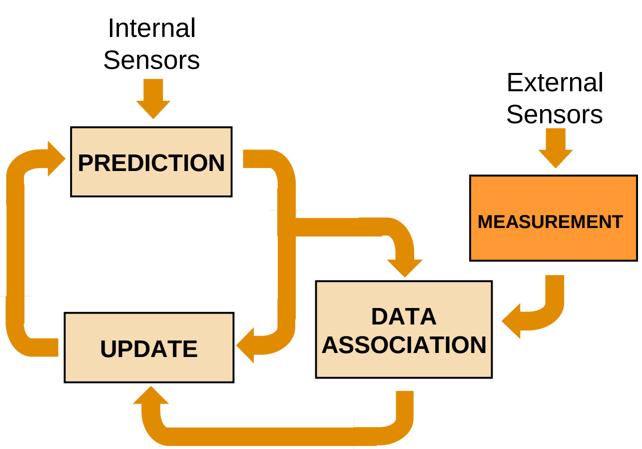
> The uncertainty of the feature decrease

> The uncertainty of the correlated features decrease

> The uncertainty of the robot decrease

> The correlations become stronger





Adding New Features

Algorithm: SLAM

$\{$- Pose initialization$\}$

$x^B_0 = \hat{x}^B_0;\ \mathrm{P}^B_0 = \hat{\mathrm{P}}^B_0;$

$\left[z_0, R_0\right] = get\_measurements;$

$\left[x^B_0, P^B_0\right] = add\_new\_features\left(x^B_0, P^B_0, z_0, R_0\right);$

for k=1 to steps **do**

$\left[u^{R_{k-1}}_{R_k}, Q_k\right] = get\_odometry$

$\{$-EKF prediction$\}$

$\left[x^B_{k|k-1}, P^B_{k|k-1}\right] = move\_vehicle\left(x^B_{k-1}, P^B_{k-1}, x^{R_{k-1}}_{R_k}, Q_k\right)$

$\left[z_k, R_k\right] = get\_measurements$

$\mathcal{H}_k = data\_association\left(x^B_{k|k-1}, P^B_{k|k-1}, z_k, R_k\right);$

$\{$-EKF update$\}$

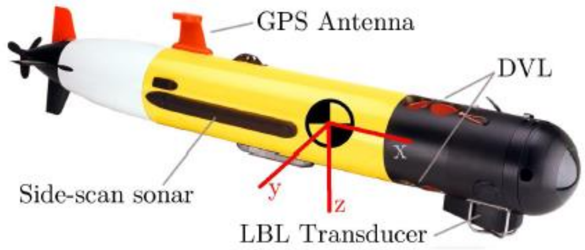
$\left[x^B_k, P^B_k\right] = update\_position\left(x^B_{k|k-1}, P^B_{k|k-1}, z_k, R_k\right);$

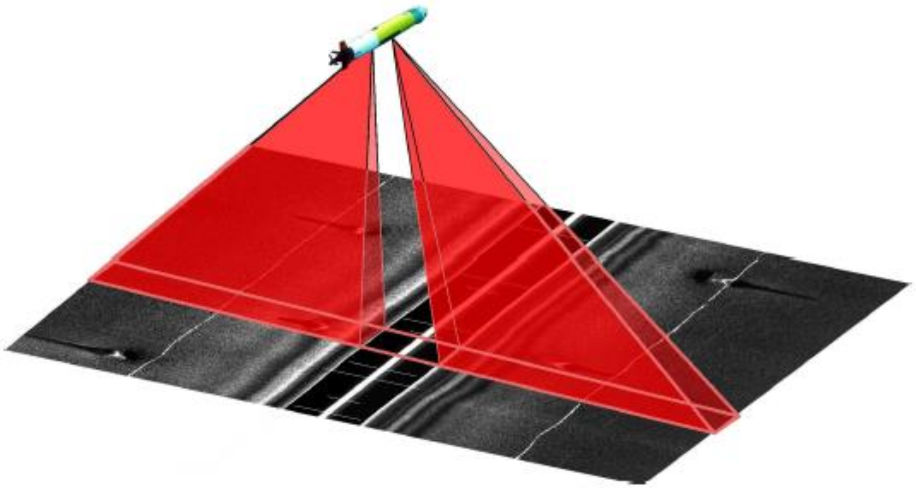
$\left[x^B_k, P^B_k\right] = add\_new\_features\left(x^B_k, P^B_k, z_k, R_k, \mathcal{H}_k\right);$

end for

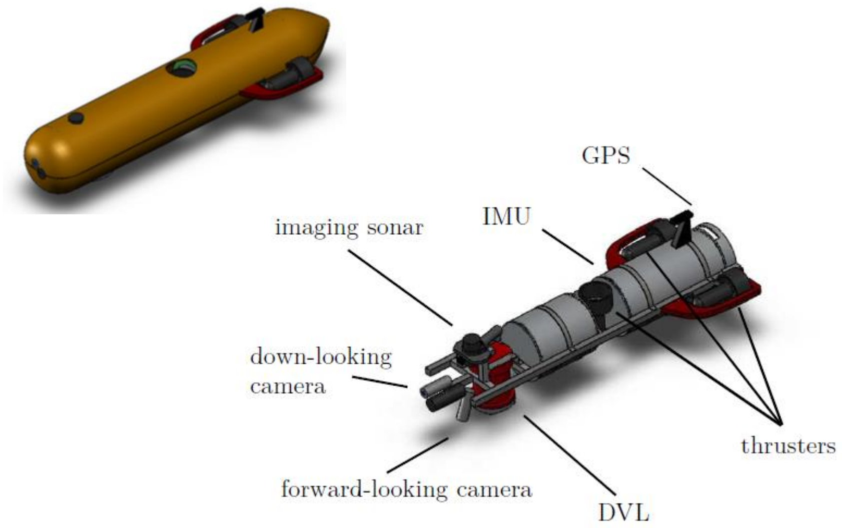
Samples

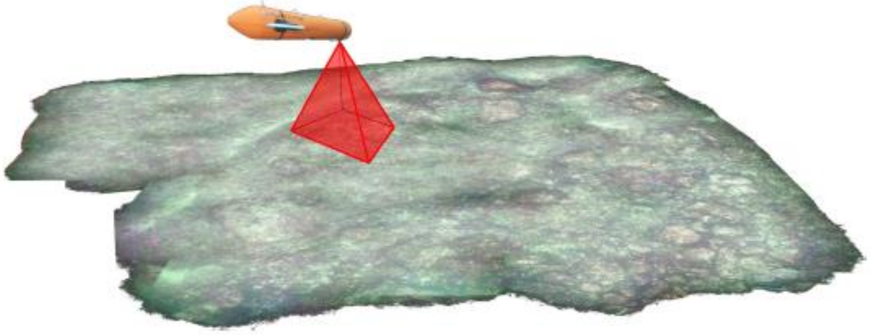
REMUS 100 AUV



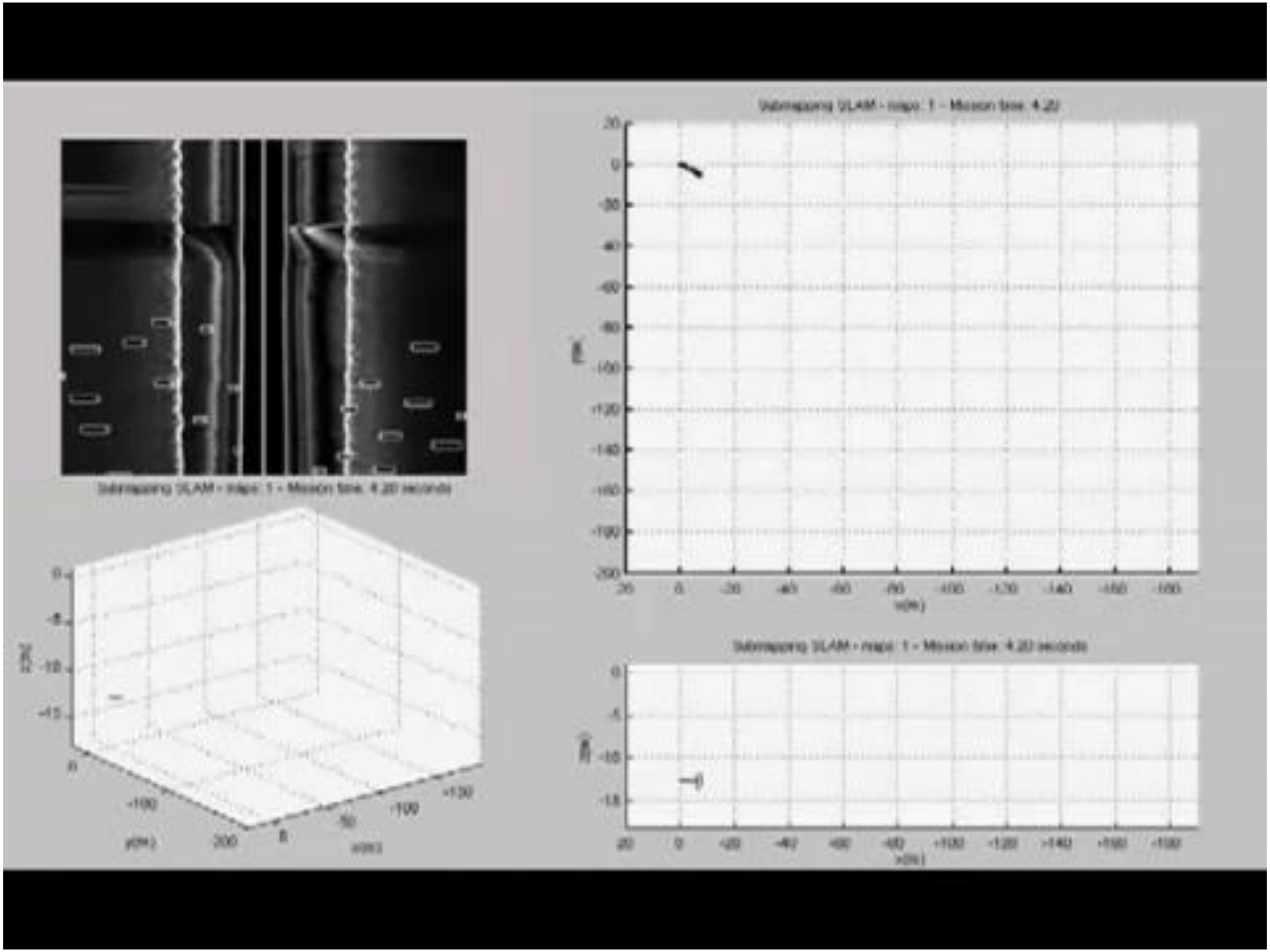


SPARUS AUV

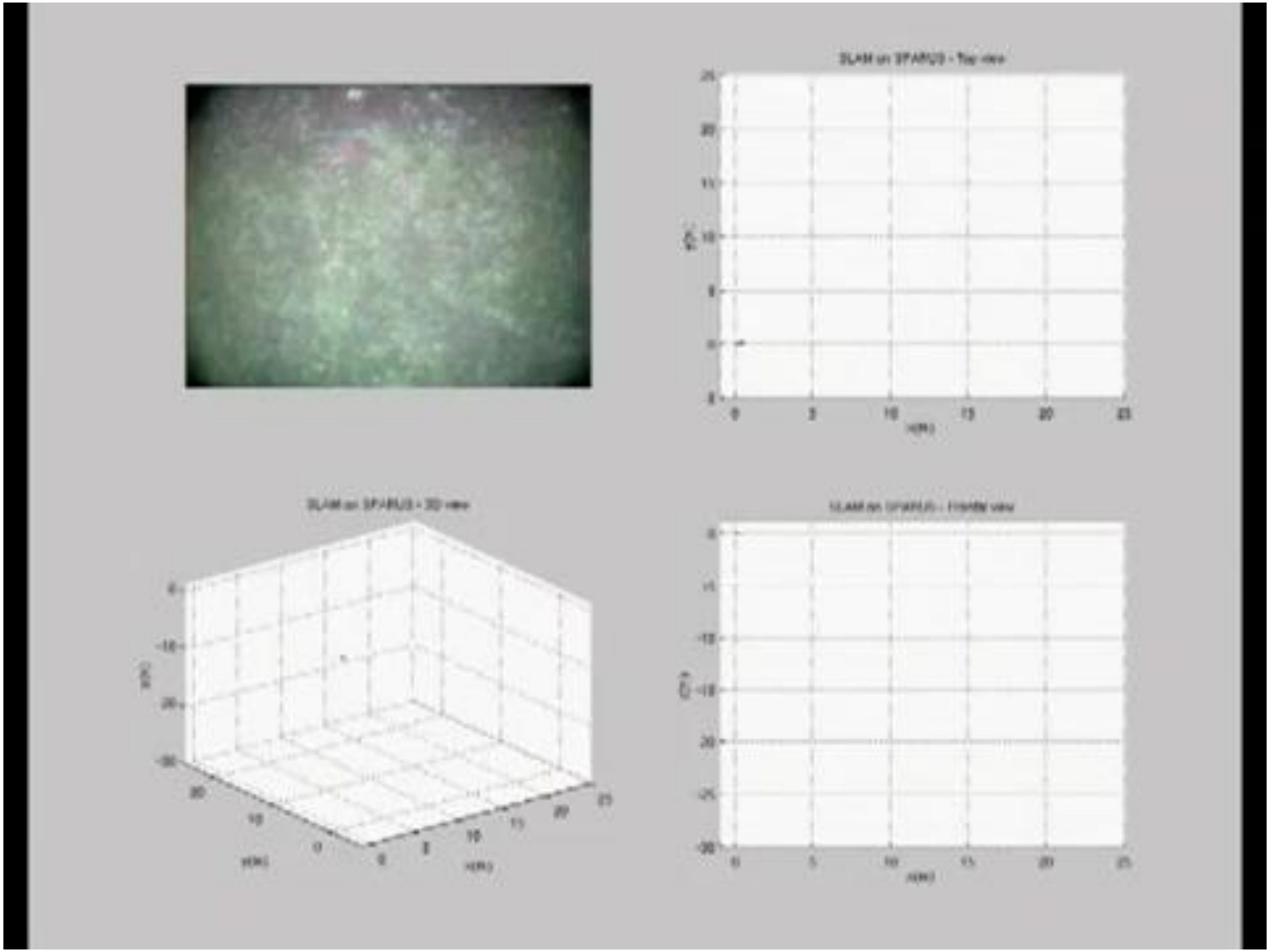




REMUS 100 AUV

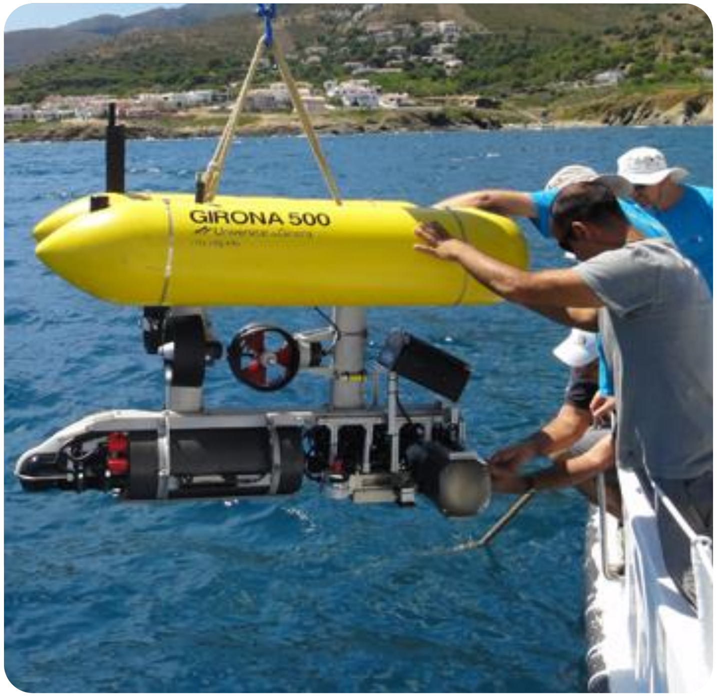


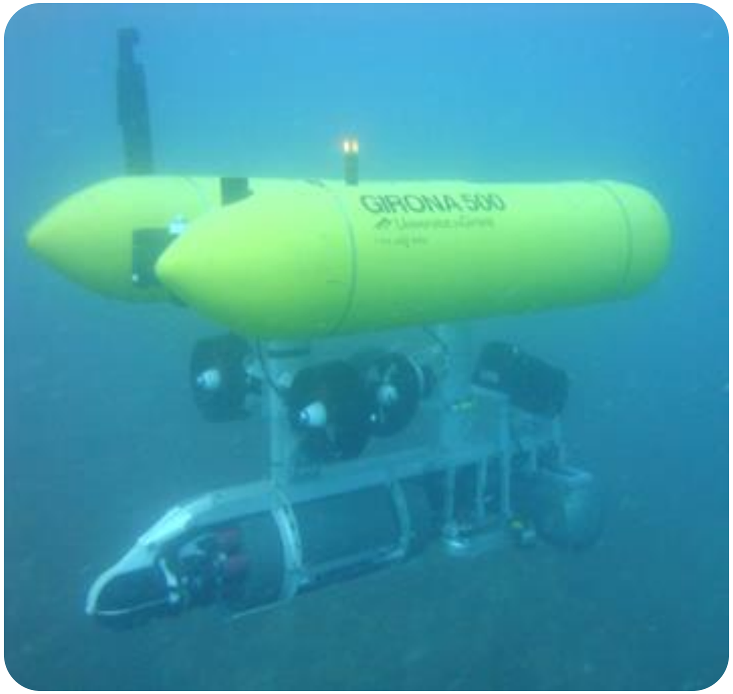
SPARUS AUV



Samples

GIRONA 500 AUV





AUTONOMOUS UNDERWATER INTERVENTION

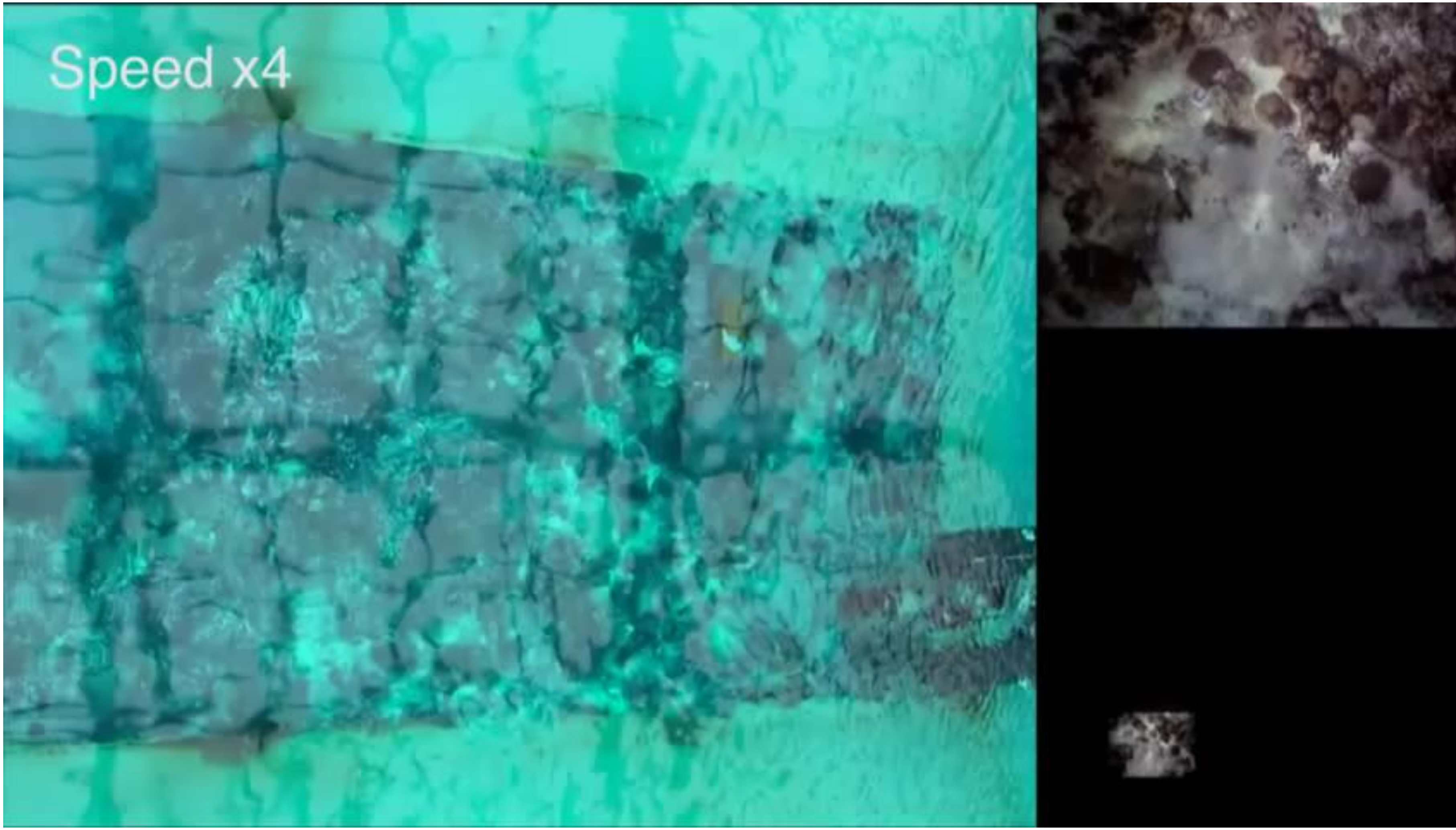
Black-Box Recovery

UNIVERSITAT JAUME·I

Universitat de Girona

Universitat de les Illes Balears

GIRONA 500 AUV



GIRONA 500 AUV