Dead Reckoning In Field Time (DRIFT)

NA 568, Winter 2024
Mobile Robotics

Tzu-Yuan (Justin) Lin
Friday, February 16, 2024

Why Proprioceptive State Estimation?

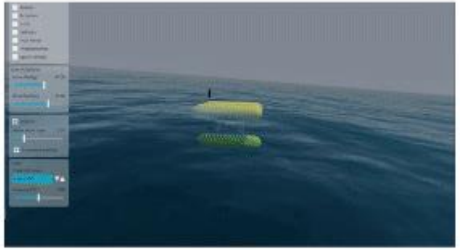
DRIFT: Dead Reckoning In Field Time [1]

Legged Robots

Full-size Vehicles

Field Robots

Indoor Robots



Marine Robots

1. Lin, Tzu-Yuan, Tingjun Li, Wenzhe Tong, and Maani Ghaffari. "Proprioceptive Invariant Robot State Estimation." arXiv preprint arXiv:2311.04320 (2023).

https://github.com/UMich-CURLY/drift

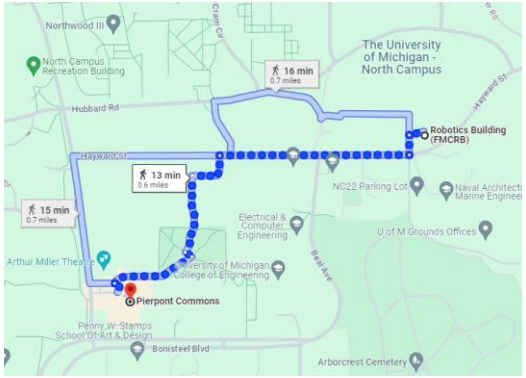
State Estimation

Local Consistency

- Only local information is needed
- High frequency update of the pose & velocity
- Odometry system

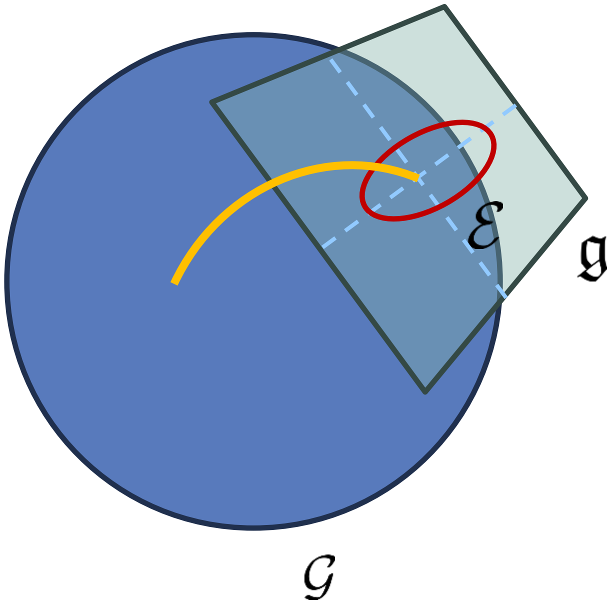
Global Consistency

- Global map for long-term planning
- Low frequency update
- SLAM with loop closure



Invariant Kalman Filtering [3]

- Means evolves on the group.
- Tracks the covariance in the Lie algebra.



3. Barrau, Axel, and Silvère Bonnabel. "The invariant extended Kalman filter as a stable observer." IEEE Transactions on Automatic Control 62.4 (2016): 1797-1812.

Invariant Kalman Filtering

Propagation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{\mathbf{X}}_t = f_{u_t}(\bar{\mathbf{X}}_t)$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{P}_t = \mathbf{A}_t\mathbf{P}_t + \mathbf{P}_t\mathbf{A}_t^{\mathsf{T}} + \bar{\mathbf{Q}}_t,$$

Correction:

$$\bar{\mathbf{X}}_t^{+} = \operatorname{Exp}\Big(\underbrace{\mathbf{K}_t\mathbf{\Pi}\left(\bar{\mathbf{X}}_t\mathbf{Y}_t\right)}_{\text{correction vector}}\Big)\,\bar{\mathbf{X}}_t$$
$$\mathbf{P}_t^{+} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t$$

$$\left.\begin{aligned}\mathbf{S}_t &= \mathbf{H}_t\mathbf{P}_t\mathbf{H}_t^{\mathsf{T}} + \bar{\mathbf{N}}_t \\ \mathbf{K}_t &= \mathbf{P}_t\mathbf{H}_t^{\mathsf{T}}\mathbf{S}_t^{-1}\end{aligned}\right\}\ \text{Computing Kalman Gain}$$

Linearization are constant!

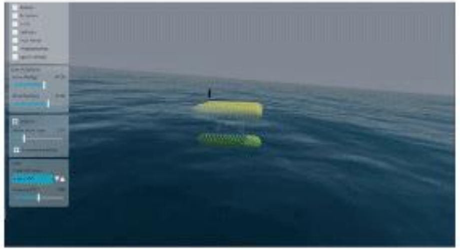
DRIFT: Dead Reckoning In Field Time

Legged Robots

Full-size Vehicles

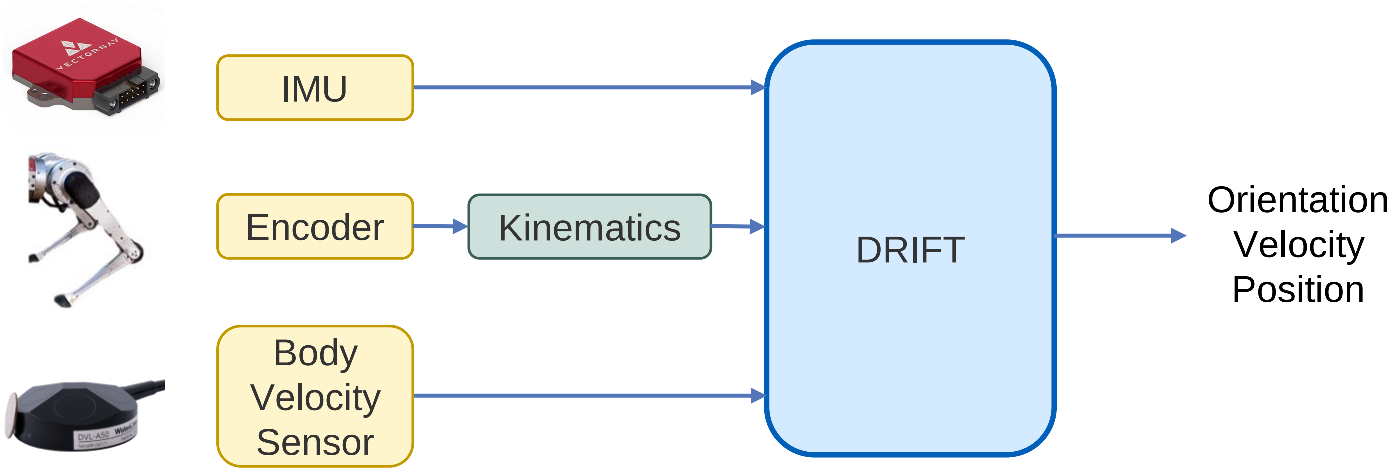
Field Robots

Indoor Robots

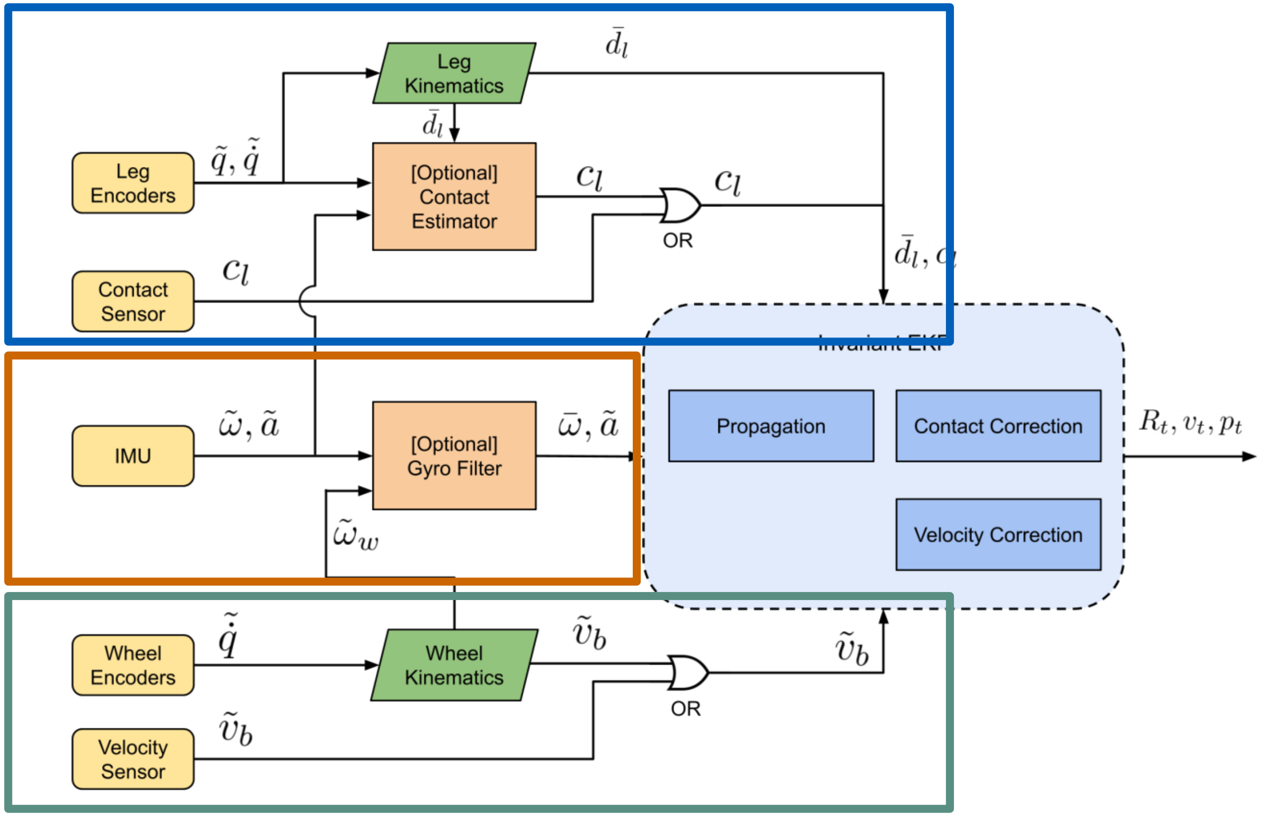


Marine Robots

Objective


IMU
Encoder
Kinematics
Body
Velocity
Sensor
DRIFT
Orientation
Velocity
Position

DRIFT: Dead Reckoning In Field Time



State Definition

$$X_t \in \mathrm{SE}_{l+2}(3)$$

$$X_t := \begin{bmatrix} R_t & v_t & p_t & d_{1t} & \cdots & d_{lt} \\ 0_{l+2,3} & & & I_{l+2} & & \end{bmatrix}$$

$R_t \in SO(3)$: Rotation Matrix

$v_t \in \mathbb{R}^3$: Velocity Vector

$p_t \in \mathbb{R}^3$: Position Vector

$d_{lt} \in \mathbb{R}^3$: Contact Position Vector

IMU Measurements

$$\tilde{\omega}_t = \omega_t + w_t^g, \quad w_t^g \sim \mathcal{GP}(0_{3,1}, \Sigma^g \delta(t - t'))$$
$$\tilde{a}_t = a_t + w_t^a, \quad w_t^a \sim \mathcal{GP}(0_{3,1}, \Sigma^a \delta(t - t'))$$

IMU Propagation – Continuous Dynamics

$$\frac{d}{dt}\underline{R_t} = \underline{R_t}(\underline{\tilde{\omega}_t - w_t^g})_\times$$

$$\frac{d}{dt}\underline{v_t} = \underline{R_t}(\underline{\tilde{a}_t - w_t^a}) + \underline{g}$$

$$\frac{d}{dt}\underline{p_t} = \underline{v_t}$$

— : World frame

— : Body frame

IMU Propagation – Continuous Dynamics

$$\frac{d}{dt}R_t = R_t(\tilde{\omega}_t - w_t^g)_\times$$

$$\frac{d}{dt}v_t = R_t(\tilde{a}_t - w_t^a) + g$$

$$\frac{d}{dt}p_t = v_t$$

$$\frac{d}{dt}X_t = \boxed{\begin{bmatrix} R_t(\tilde{\omega}_t)_\times & R_t\tilde{a}_t + g & v_t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

$$- \boxed{\begin{bmatrix} R_t & v_t & p_t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \boxed{\begin{bmatrix} (w_t^g)_\times & w_t^a & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}$$

$$:= \underbrace{\boxed{f_{u_t}(X_t)}}_{\text{Deterministic Dynamics}} - \underbrace{\boxed{X_t}\boxed{w_t^\wedge}}_{\text{Noise term}}$$

IMU Propagation – Continuous Dynamics

$$\boxed{f_{u_t}(X_t)} = \boxed{\begin{bmatrix} R_t(\tilde{\omega}_t)_\times & R_t\tilde{a}_t + g & v_t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

Check

$$\boxed{\begin{aligned} & f_{u_t}(X_1 X_2) \\ & = f_{u_t}(X_1)X_2 + X_1 f_{u_t}(X_2) - X_1 f_{u_t}(I) X_2 \end{aligned}}$$

OK!

Is group affine!

The error dynamics

$$\eta_t^r = \bar{X}_t X_t^{-1}$$

$$\frac{d}{dt}\eta_t^r = \underbrace{f_{u_t}(\eta_t^r) - \eta_t^r f_{u_t}(I)}_{\text{Deterministic term} := g_{u_t}(\eta_t^r)} + \underbrace{(\bar{X} w_t^\wedge \bar{X}^{-1})\eta_t^r}_{\text{Noise term}}$$

Deterministic term $:= g_{u_t}(\eta_t^r)$

Noise term

IMU Propagation – Error Dynamics

$$\eta_t^r = \bar{X}_t X_t^{-1}$$

The error dynamics

$$\frac{d}{dt}\eta_t^r = f_{u_t}(\eta_t^r) - \eta_t^r f_{u_t}(I) + (\bar{X} w_t^\wedge \bar{X}^{-1})\eta_t^r$$

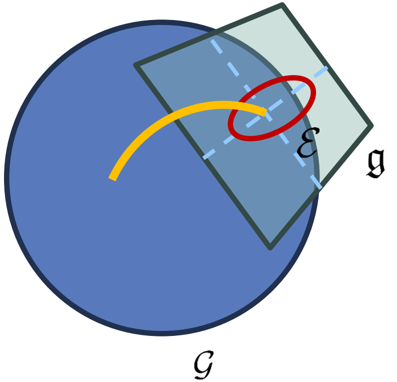
On the group

We want to track the error in the Lie algebra!

$$\eta_t^r = \exp(\xi_t)$$

$$\frac{d}{dt}\xi_t^r = A_t \xi_t^r + Ad_{\bar{X}_t}\xi_t^r$$

Error dynamics in the Lie algebra



We need to find A_t^r!!

$$f_{u_t}(X_t) = \begin{bmatrix} R_t(\tilde{\omega}_t)_\times & R_t\tilde{a}_t + g & v_t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Longrightarrow A_t^r = \begin{bmatrix} 0 & 0 & 0 \\ (g)_\times & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$

Linearization is constant!!

IMU Propagation – Discrete Integration

Mean Propagate through discrete integration

Ex.

$$\frac{d}{dt}R_t = R_t(\tilde{\omega}_t - w_t^g)_\times \implies \bar{R}_{t_{k+1}} = \bar{R}_{t_k} \exp\left(\omega_{t_k}\Delta t\right)$$

Covariance Propagate via the state transition matrix

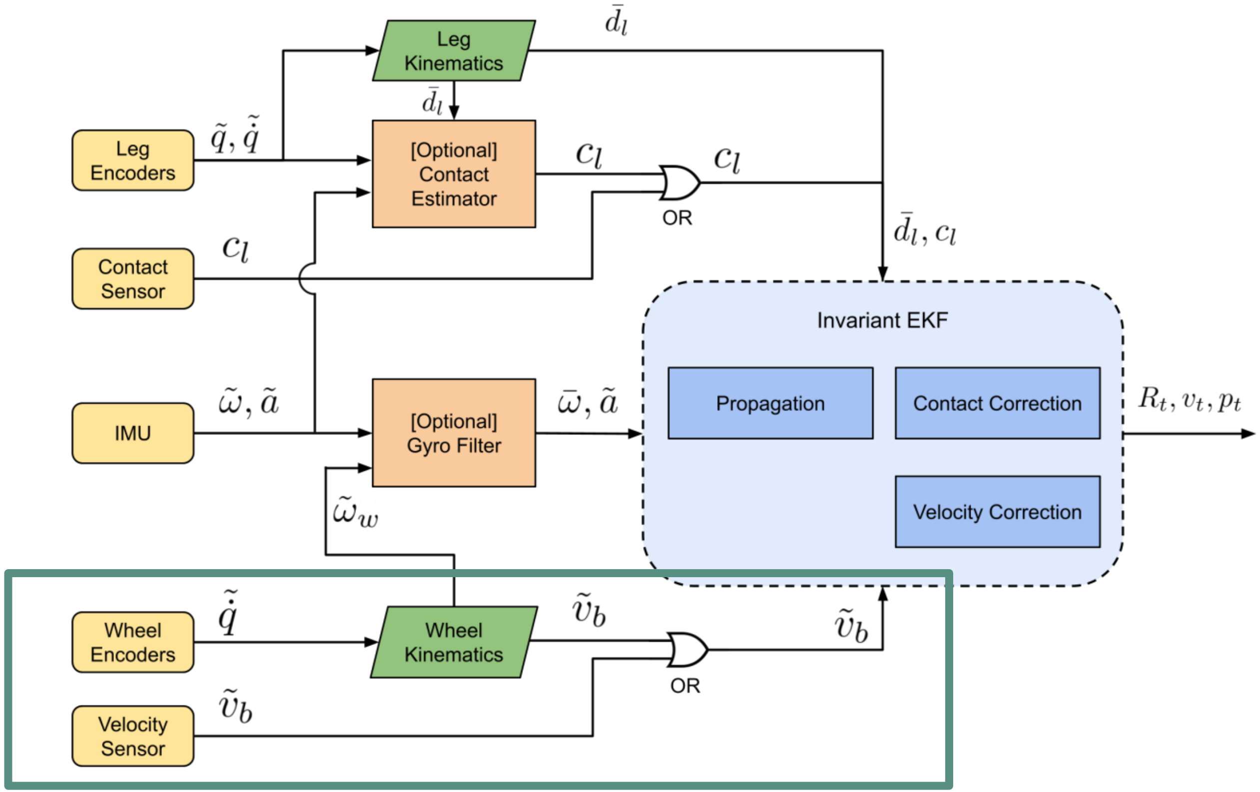
State transition matrix in the Lie algebra

State transition matrix on the group

$$\Phi^r = \exp(A^r \Delta t)$$

$$P_{k+1} = \Phi^r P_k \Phi^{r\top} + \mathrm{Ad}_{\bar{X}_k} Q_d \mathrm{Ad}_{\bar{X}_k}^\top$$

DRIFT: Dead Reckoning In Field Time



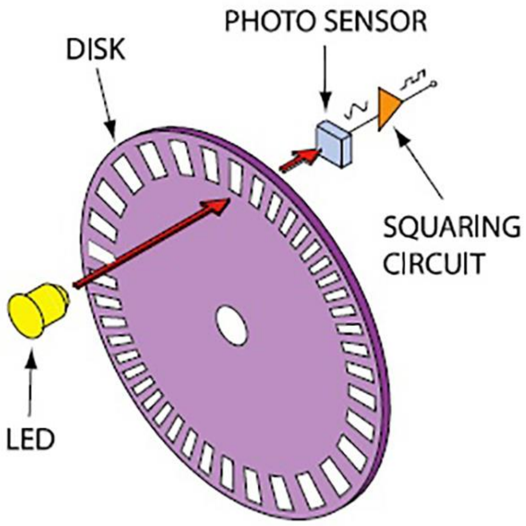
Vehicles & Wheeled Robots

Full-size Vehicles

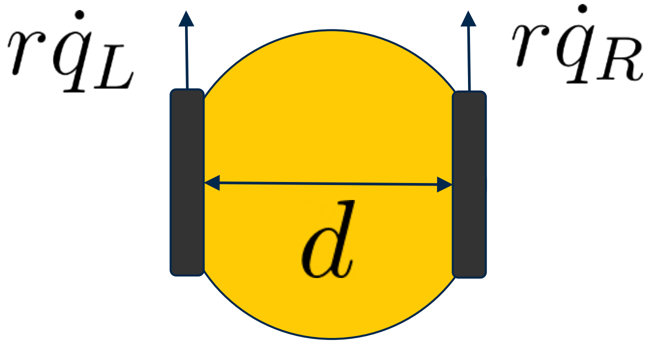
Field Robots

Indoor Robots

Encoders



$$v_w = \begin{bmatrix} \frac{r(\dot{q}_R+\dot{q}_L)}{2} & 0 & 0 \end{bmatrix}^\top$$



Velocity Correction

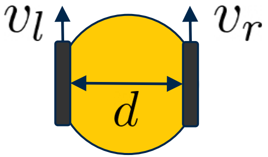

Full-size Vehicles


Field Robots


Indoor Robots

$$Y_{t_k} = X_{t_k}^{-1} b + V_{t_k}$$

$$\begin{bmatrix} \tilde{v}_{t_k} \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} R_{t_k}^{\top} & -R_{t_k}^{\top} v_{t_k} & -R_{t_k}^{\top} p_{t_k} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} w_{t_k}^{v} \\ 0 \\ 0 \end{bmatrix}$$



Velocity Correction


Full-size Vehicles


Field Robots


Indoor Robots

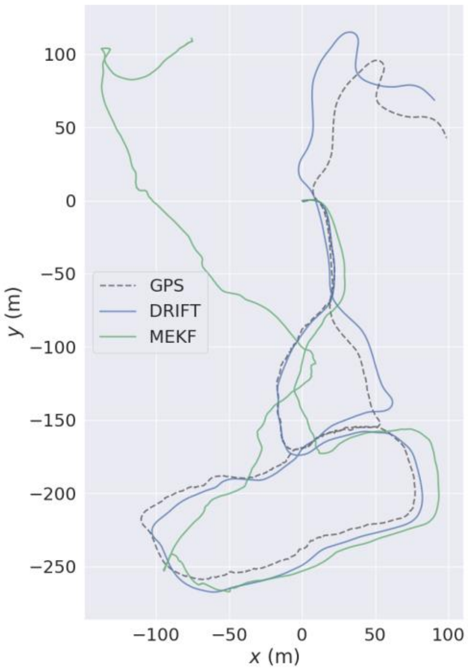
$$H\xi_k^r = -\xi_k^{r\wedge} b$$

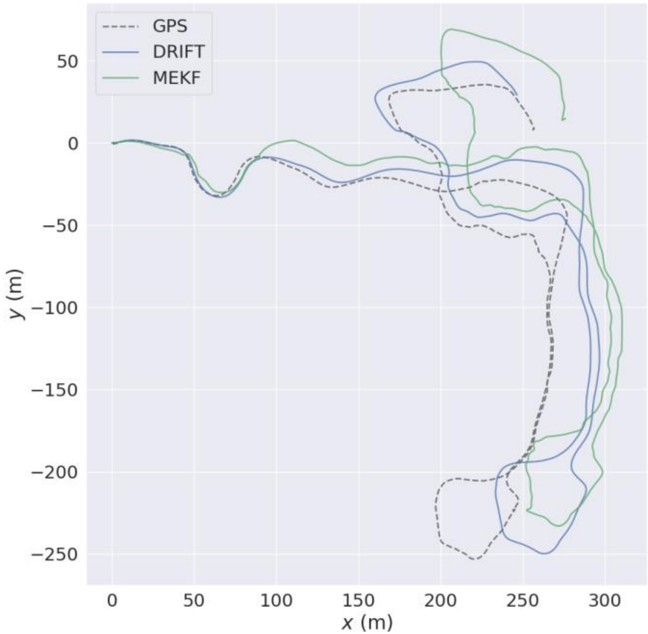
$$H \begin{bmatrix} \xi_k^\omega \\ \xi_k^v \\ \xi_k^p \end{bmatrix} = - \begin{bmatrix} \xi_k^{\omega\wedge} & \xi_k^v & \xi_k^p \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} \xi^v \\ 0 \\ 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 0_{1,3} & I & 0 \\ 0_{1,3} & 0 & 0 \\ 0_{1,3} & 0 & 0 \end{bmatrix}$$

Full-size Vehicles

Full-Size Vehicle





- 3 Sequences
- Avg. Distance: 1510.43 m
- Avg. Duration: 449.15 sec

	MEKF [5]	DRIFT
Final Drift (m)	203.02	**51.08**
Percentage (%)	12.32%	**3.18%**

5. Sola, Joan. "Quaternion kinematics for the error-state Kalman filter." *arXiv preprint arXiv:1711.02508* (2017).

Field Robots

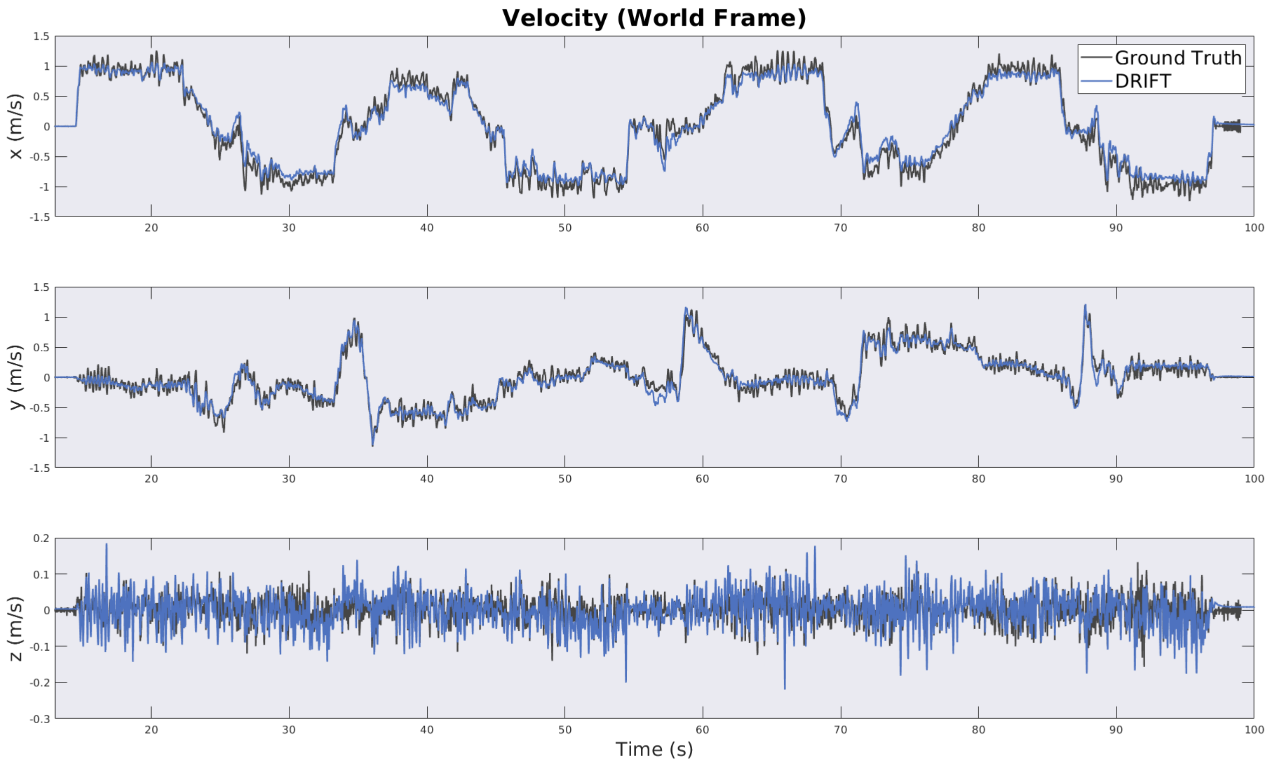
Experiment with a Motion Capture System

- 10 Sequences
- Avg. Distance: 49.17 m
- Avg. Duration: 85.10 sec

Relative Pose Error	MEKF [5]	DRIFT
Trans. (m/m)	0.0747	**0.0701**
Rot. (°/m)	2.0485	**1.6888**

5. Sola, Joan. "Quaternion kinematics for the error-state Kalman filter." *arXiv preprint arXiv:1711.02508* (2017).

Velocity Estimation



Indoor Robot

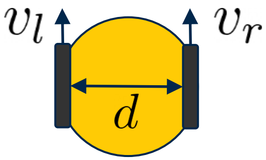
Yaw angle is not observable

Low-cost IMUs can produce disastrous result due to the time-varying biases

One extra information that is not used:

Angular velocity from kinematics!

$$\hat{\omega}_z = \frac{v_r - v_l}{d}$$



Gyro Filter

State

$$x := \begin{bmatrix} \omega^\top & b^{g\top} \end{bmatrix}^\top$$

Propagation

Assume same bias between two measurements

$$x_{k+1} = x_k + \begin{bmatrix} \underbrace{\tilde{\omega}^{\alpha}_{k+1} - \tilde{\omega}^{\alpha}_{k}}, \\ 0_{3\times 1} \end{bmatrix}$$

Correction

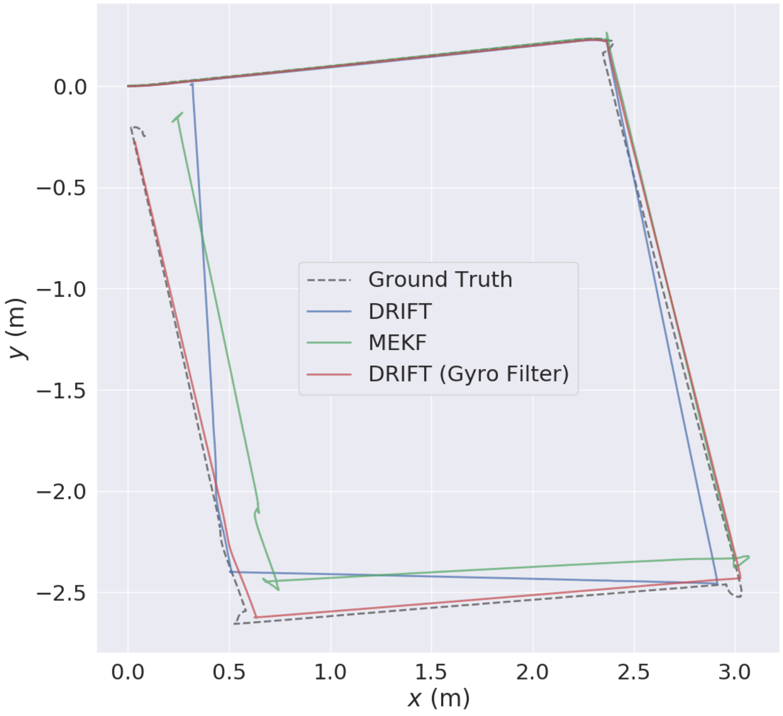
$$\tilde{\omega}^{\beta} = Hx$$

$$H = \begin{bmatrix} I_{3\times 3} & I_{3\times 3} \end{bmatrix} \text{ or } \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} \end{bmatrix}$$

Biased Unbiased

Indoor Robots

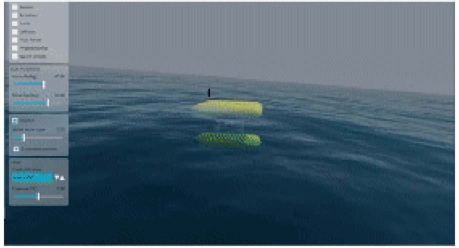
Motion Capture Experiment



- 8 indoor sequences

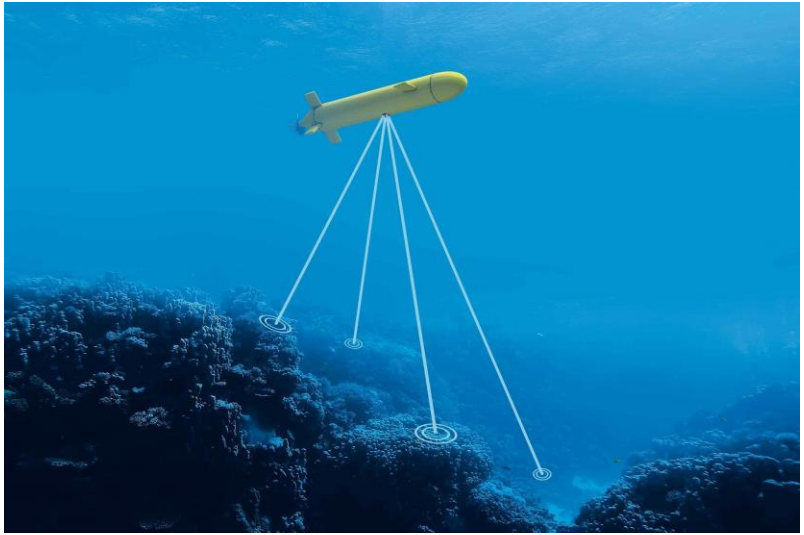
Relative Pose Error	MEKF	DRIFT	DRIFT (Gyro Filter)
Trans. (m/m)	0.0844	0.0692	**0.0590**
Rot. (°/m)	3.6460	3.6198	**3.5631**

Marine Robots



Marine Robots

Marine Robots

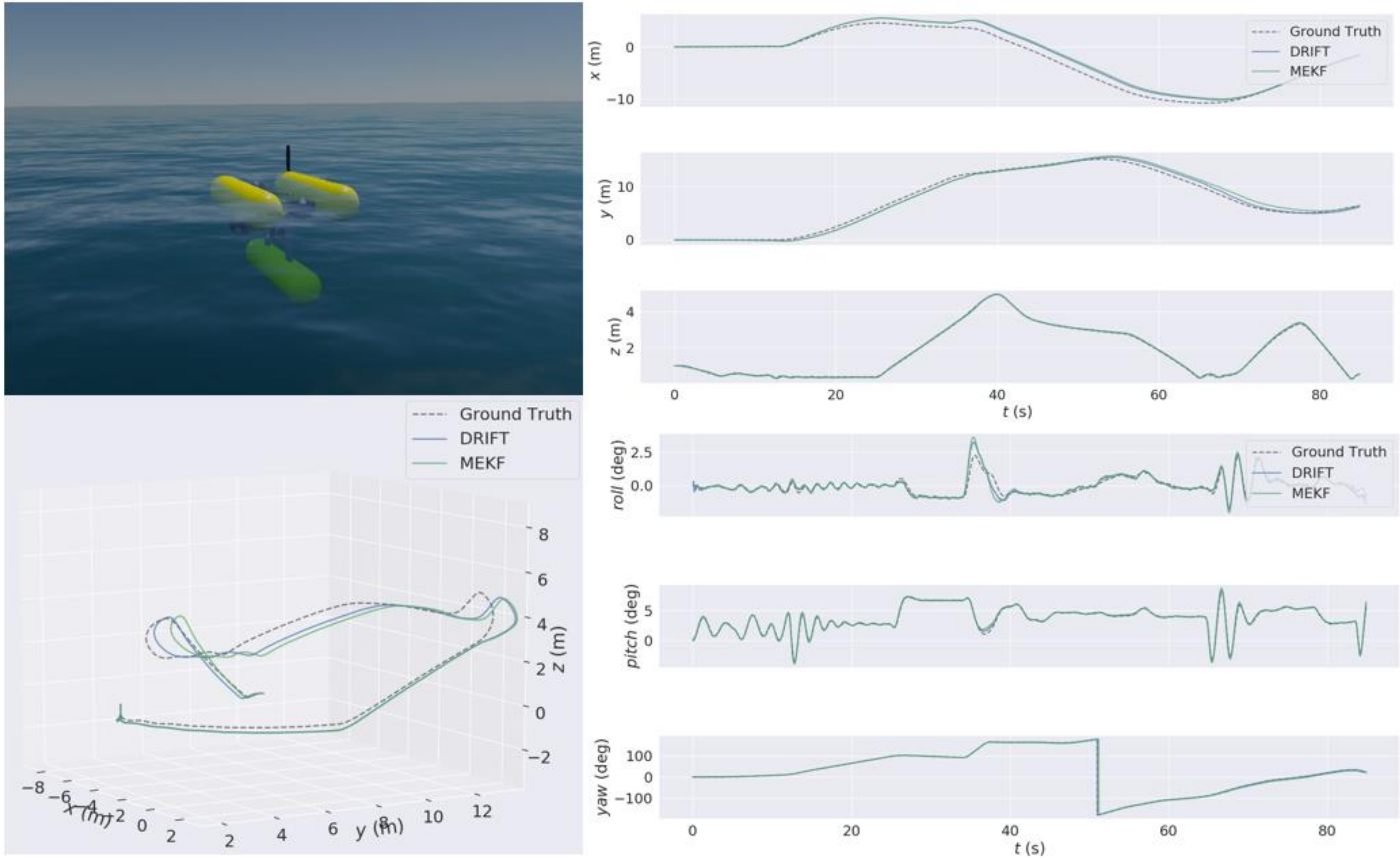


Doppler Velocity Logs (DVL)

- seabed-referenced body velocity
- Acoustic beams + the Doppler effect

Marine Robots

Marine Robots



Legged Robots

Legged Robots

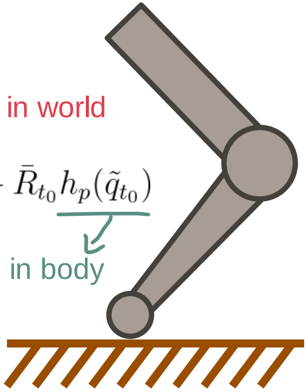
Contact Augmentation [4]

Foot position in the world does not change!

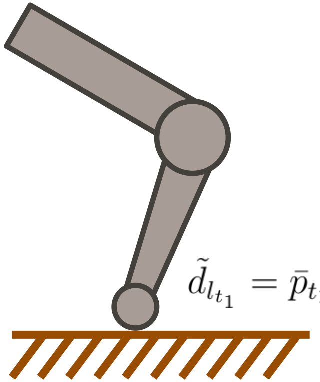
Foot position in world

$$\tilde{d}_{l_{t_0}} = \bar{p}_{t_0} + \bar{R}_{t_0} h_p(\tilde{q}_{t_0})$$

Foot position in body (kinematics)

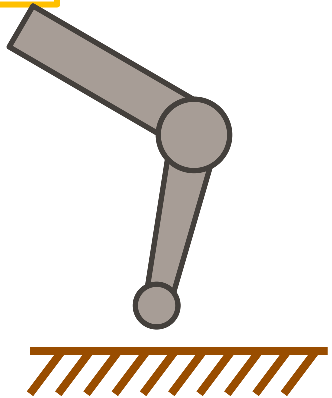


First Contact



$$\tilde{d}_{l_{t_1}} = \bar{p}_{t_1} + \bar{R}_{t_1} h_p(\tilde{q}_{t_1})$$

Body Moves Forward



Foot Lift Off

$$X_t = \begin{bmatrix} R_t & v_t & p_t & \boxed{\begin{matrix} d_{lt_0} \\ 0 \\ 0 \\ 1 \end{matrix}} \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \end{bmatrix}$$

Add!

Correction Step:

$\tilde{d}_{lt_1}$ should match $\tilde{d}_{lt_0}$

$$X_t = \begin{bmatrix} R_t & v_t & p_t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Remove!

4. Hartley, R., Ghaffari, M., Eustice, R.M. and Grizzle, J.W., 2020. Contact-aided invariant extended Kalman filtering for robot state estimation. The International Journal of Robotics Research, 39(4), pp.402-430.

Contact Propagation [4]

$$X_t = \begin{bmatrix} R_t & v_t & p_t & d_t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Legged Robots

Foot position doesn't change during the contact period

Only affected by the white noise

$$\frac{d}{dt} d_t = R_t h_R(\tilde{q}_t) \underbrace{(-w_t^d)}_{\text{Noise}}$$

4. Hartley, R., Ghaffari, M., Eustice, R.M. and Grizzle, J.W., 2020. Contact-aided invariant extended Kalman filtering for robot state estimation. The International Journal of Robotics Research, 39(4), pp.402-430.

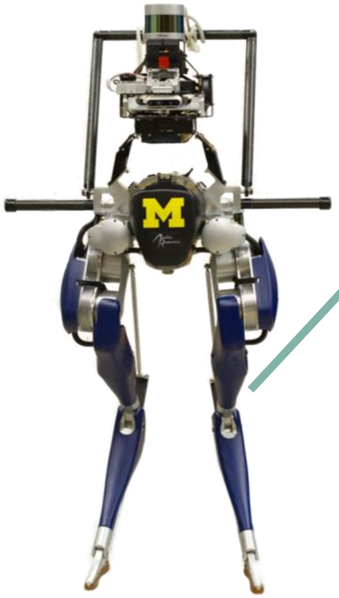
Contact Correction [4]

$$Y_{t_k} = X_{t_k}^{-1} b + V_{t_k}$$

$$\begin{bmatrix} h_p(\tilde{q}_{t_k}) \\ 0 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} R_{t_k}^\top & -R_{t_k}^\top v_{t_k} & -R_{t_k}^\top p_{t_k} & -R_{t_k}^\top d_{t_k} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} + \begin{bmatrix} J_p(\tilde{q}_{t_k}) w_{t_k}^q \\ 0 \\ 0 \end{bmatrix}$$

4. Hartley, R., Ghaffari, M., Eustice, R.M. and Grizzle, J.W., 2020. Contact-aided invariant extended Kalman filtering for robot state estimation. The International Journal of Robotics Research, 39(4), pp.402-430.

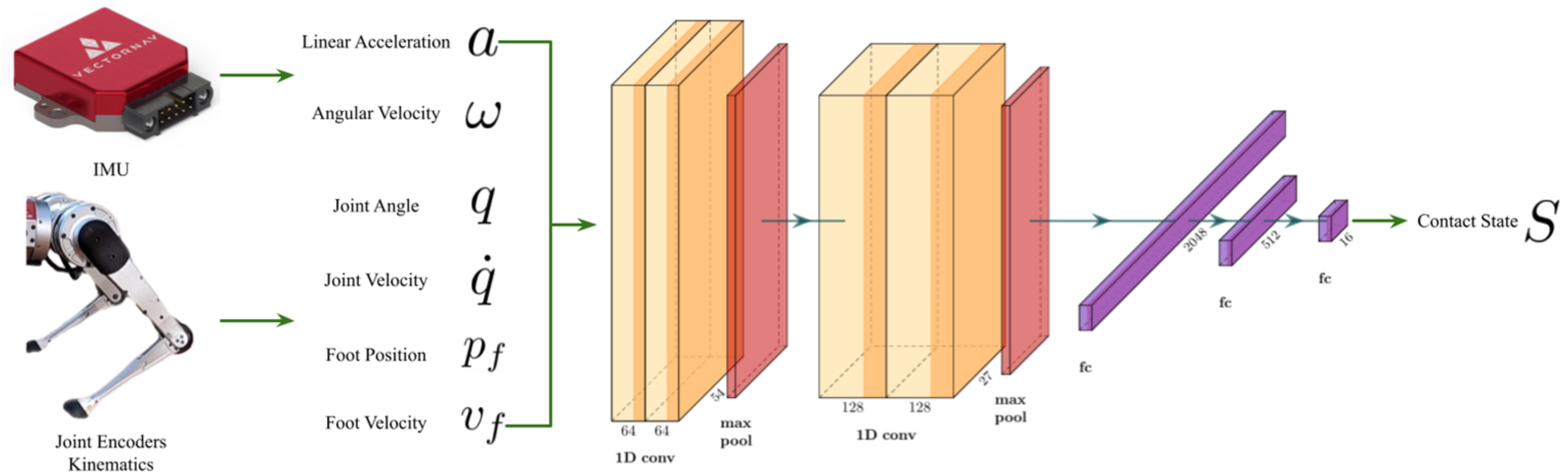
Legged Robot - Contact Detection



Spring for contact detection!

How do we accurately determine contact events?

Deep Contact Estimator



Runs real-time on an NVIDIA Jetson AGX Xavier at 830 Hz!

Contact Estimation Results

Accuracy	Concrete Test Set	Grass Test Set	Forest Test Set
GRF	71.30%	82.14%	81.77%
Gait Cycle	85.11%	91.59%	83.58%
Contact Estimator	**98.18%**	**97.78%**	**97.08%**

Legged Robot

Run-Time Analysis

DRIFT runs real-time using a CPU on the robot!

	i5-11400H		AGX Xavier (CPU)	
Unit: μs	mean	std	mean	std
InEKF				
propagation	11.33	4.00	18.35	4.19
propagation with contact	10.32	4.76	22.56	7.21
velocity correction	9.91	4.80	18.46	6.66
contact correction	17.46	9.78	29.39	13.07
Gyro Filter				
propagation	2.57	3.46	3.96	2.28
correction	2.85	2.89	4.64	4.40

https://github.com/UMich-CURLY/drift

Questions?

Feel free send an email to me! :)

Tzu-Yuan (Justin) Lin

tzuyuan@umich.edu

https://github.com/UMich-CURLY/drift