Contact-aided Invariant Extended Kalman Filtering for Legged Robot State Estimation

Ross Hartley
Why do we need legged robots?

- All terrain access
- Delivery and home robots
- Inspection
- Search and rescue
What states need to be estimated?

- position
- orientation
- velocity
- joint positions/velocities
- contact states
What states need to be estimated?

- position
- orientation
- velocity
- joint positions/velocities
- contact states

Visual-inertial odometry?

- encoders
- contact sensors
Failures of visual-inertial odometry

Vision may fail when ...

• Scarcity of features
  - snow, grass ...

• Poor lighting
  - sun glare, night ...

• Obstructions
  - smoke, water on lens, ...

• Dynamic environments
• Motion blur
Failures of visual-inertial odometry

Integration of IMU measurements alone leads to significant drift due to sensor noise and bias!

VectorNav-100 IMU

<table>
<thead>
<tr>
<th>Grade</th>
<th>Accelerometer Bias Error [mg]</th>
<th>Horizontal Position Error [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1s</td>
</tr>
<tr>
<td>Navigation</td>
<td>0.025</td>
<td>0.13 mm</td>
</tr>
<tr>
<td>Tactical</td>
<td>0.3</td>
<td>1.5 mm</td>
</tr>
<tr>
<td>Industrial</td>
<td>3</td>
<td>15 mm</td>
</tr>
<tr>
<td>Automotive</td>
<td>125</td>
<td>620 mm</td>
</tr>
</tbody>
</table>

Contact implies that the stance foot remains fixed.

Forward kinematics is used to measure the foot position relative to the IMU.

Together these measurements can be used to estimate relative movement.

EKF to fuse with inertial data.

[Bloesch 2008]
Dual-estimator approach

We want stability and autonomy!

- Only local map information is needed
- High-frequency orientation/velocity for control

- Global map and pose needed for long-term planning
- Low-frequency estimate of pose is fine

Invariant-EKF fusing inertial, encoder, and contact measurements (proprioceptive only)

Factor graph smoother that fuses all measurements including vision-based loop closures (SLAM)
Types of Kalman Filters

- **Kalman Filter**: optimal linear filter – state assumed to be a Gaussian random variable
  \[ \frac{d}{dt} x_t = A_t x_t + B_t u_t + w_t \]

- **(Direct) Extended Kalman Filter (EKF)**: linearize nonlinear state dynamics (filter states directly)
  \[ \frac{d}{dt} x_t = f(x_t, u_t, w_t) \quad A_t = \frac{\partial f}{\partial x_t} \bigg|_{x = \hat{x}_t} \]

- **(Indirect or Error-State) EKF**: linearize nonlinear error dynamics (filter error states)
  \[ e_t \triangleq x_t - \hat{x}_t \quad \frac{d}{dt} e_t = g(e_t, x_t, u_t, w_t) \]
  \[ \approx A_t(\bar{x}_t, u_t) e_t + \bar{w}_t \]

In general, the linearization depends on the current state estimate.

**Bad estimate** → Incorrect linearization → Poor performance
We can choose the error-state!

- **(Indirect or Error-State) EKF**: linearize nonlinear error dynamics (filter error states)

\[
\begin{align*}
e_t & \triangleq x_t - \hat{x}_t \\
\frac{d}{dt} e_t &= g(e_t, x_t, u_t, w_t) \\
& \approx A_t(x_t, u_t) e_t + w_t
\end{align*}
\]

Different choices of error variables lead to different linearized error dynamics

- e.g. Euler angle error vs. quaternion error
Is there a choice of error variables that leads to autonomous dynamics?
(independent of the state estimate)

Yes! (group-affine systems)  Invariant Extended Kalman Filter
[Barrau 2014]
A Lie group, $\mathcal{G}$, is a group that is also a differentiable manifold.
- Examples: $\text{SO}(3)$, $\text{SE}(3)$ are matrix Lie groups

The Lie algebra is defined as the tangent space at the identity element of the group $\mathcal{T}_e\mathcal{G}$
- This vector space is isomorphic to $\mathbb{R}^n$

The group’s **exponential map**, $\exp: \mathcal{T}_e\mathcal{G} \to \mathcal{G}$, maps a vector in the Lie algebra to the Lie group
- Inverse is the **logarithm map**, $\log: \mathcal{G} \to \mathcal{T}_e\mathcal{G}$

“hat operator”
$$(\cdot)^\wedge: \mathbb{R}^n \to \mathcal{T}_e\mathcal{G}$$

Vectorized notation:
$$\text{Exp}(\xi) \triangleq \exp (\xi^\wedge)$$
$$\text{Log} (\text{Exp}(\xi)) = \xi$$

\[\begin{align*}
\text{Exp}(\xi_t) &= \eta_t \\
\xi_t &\in \mathbb{R}^n \\
\eta_t &\in \mathcal{G}
\end{align*}\]
Invariant Kalman Filtering

- System defined on a **matrix Lie group**: $X_t \in \mathcal{G} \quad \text{SO}(3), \text{SE}(3), \text{etc.}$

- Dynamics satisfy **“group affine” property**: $f_u(t)(X_1X_2) = f_u(t)(X_1)X_2 + X_1f_u(t)(X_2) - X_1f_u(t)(I_d)X_2$

**With “group affine” systems, a particular choice of error variables will lead to log-linear error dynamics.**

- Error is defined through matrix multiplication:
  
  $$\eta_t^r = X_tX_t^{-1} \quad \text{(Right-Invariant Error)}$$
  $$\eta_t^l = X_t^{-1}X_t \quad \text{(Left-Invariant Error)}$$

- Error is invariant to (right or left) actions of the group
  
  $$\left(LX_t\right)^{-1}L\bar{X}_t = X_t^{-1}L^{-1}L\bar{X}_t = X_t^{-1}\bar{X}_t$$

True State  Estimate

[Barrau and Bonnabel 2017]
Log-Linear Error Dynamics

The nonlinear error dynamics is exactly determined by a linear system in the Lie algebra!!!
Invariant Observations

\[ Y_t = X_t^{-1} b + V_t \]  (Right Invariant Observation)
\[ Y_t = X_t b + V_t \]  (Left Invariant Observation)

measurement = state * constant + noise

Both the linearized dynamics and measurement models will be state independent!
Contact-Aided Invariant EKF

Propagation

• Use IMU measurements to predict base frame movement.

• Use contact sensor measurement to predict supporting feet movement (zero translation).

Correction

• Use encoder measurements and forward kinematics to correct state estimate.

[ Hartley et al. RSS 2018]
[ Hartley et al. IJRR 2019]
States and Inputs

• The state is expressed as a matrix Lie group, \( X_t \in SE_K(3) \)

\[
\begin{bmatrix}
R_{WB}(t) & wv_{WB}(t) & wp_{WB}(t) \\
0_{1,3} & 1 & 0 \\
0_{1,3} & 0 & 1 \\
0_{1,3} & 0 & 0 \\
\vdots & \vdots & \vdots \\
0_{1,3} & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
wv_{WC_1}(t) & \cdots & wp_{WC_N}(t) \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1 \\
\end{bmatrix}
\]

\[X_t \triangleq \begin{bmatrix}
R_{t} & v_t & p_t & d_t \\
0_{1,3} & 1 & 0 & 0 \\
0_{1,3} & 0 & 1 & 0 \\
0_{1,3} & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[u_t \triangleq \begin{bmatrix}
B\tilde{\omega}_{WB}(t) \\
B\tilde{a}_{WB}(t)
\end{bmatrix} \triangleq \begin{bmatrix}
\tilde{\omega}_t \\
\tilde{a}_t
\end{bmatrix}
\]

accelerometer and gyroscope measurements

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
The “strapdown” inertial-contact model circumvents using the full dynamics.

\[
\dot{R}_t = R_t (\tilde{\omega}_t - w_t^q) \times \\
\dot{v}_t = R_t (\tilde{\alpha}_t - w_t^a) + g \\
\dot{p}_t = v_t \\
\dot{d}_t = R_t R_{BC} (\tilde{\alpha}_t) (-w_t^\nu)
\]

orientation of contact frame with respect to body frame

Do we have to use to robot’s complicated dynamics? No!

[Hartley et al. RSS 2018]  
[Hartley et al. IJRR 2019]
Inertial-Contact Dynamics Model

The “strapdown” inertial-contact model circumvents using the full dynamics

\[
\begin{align*}
\dot{\mathbf{q}} &= \mathbf{C}(q, \dot{q}) \dot{q} + \mathbf{G}(q) - \mathbf{B} u + \mathbf{J}^T(q) \mathbf{F} \\
\dot{\mathbf{v}}_t &= \mathbf{R}_t (\mathbf{\tilde{a}}_t - \mathbf{w}_t^g) + \mathbf{g} \\
\dot{\mathbf{p}}_t &= \mathbf{v}_t \\
\dot{\mathbf{d}}_t &= \mathbf{R}_t \mathbf{R}_{BC}(\mathbf{\tilde{a}}_t) (-\mathbf{w}_t^v)
\end{align*}
\]

orientation of contact frame with respect to body frame

Written in matrix form:

\[
\frac{d}{dt} \mathbf{X}_t = \begin{bmatrix}
\mathbf{R}_t (\mathbf{\dot{\omega}}_t) \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} + \begin{bmatrix}
\mathbf{R}_t \mathbf{\tilde{a}}_t + \mathbf{g} \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \mathbf{v}_t \begin{bmatrix}
\mathbf{0}_{3 \times 1}
\end{bmatrix}
\]

\[
- \begin{bmatrix}
\mathbf{R}_t \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_t \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
\mathbf{p}_t \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
\mathbf{d}_t \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
(\mathbf{w}_t^g) \times \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
\mathbf{w}_t^g \\
\mathbf{0}_{1 \times 3}
\end{bmatrix} \begin{bmatrix}
\mathbf{0}_{3 \times 1} \\
\mathbf{R}_{BC}(\mathbf{\tilde{a}}_t) \mathbf{w}_t^v
\end{bmatrix}
\]

\[
\triangleq \mathbf{f}_{\delta}(\mathbf{X}_t) - \mathbf{X}_t (\mathbf{w}_t)^\wedge
\]

Satisfies group affine property!

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Log-Linear Error Dynamics

(Right Invariant Error)

\[ \eta_t^r \triangleq \bar{X}_t X_t^{-1} = \text{Exp}(\xi_t) \]

\[ A_t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ (g) \times & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ \frac{d}{dt} \xi_t = A_t \xi_t + \text{Ad}_\bar{X} w_t \]

Linearized error dynamics matrix is independent of the state estimate!

(Left Invariant Error)

\[ \eta_t^l \triangleq X_t^{-1} \bar{X}_t = \text{Exp}(\xi_t) \]

\[ A_t = \begin{bmatrix} - (\hat{\omega}_t) \times & 0 & 0 & 0 \\ - (\hat{a}_t) \times & - (\hat{\omega}_t) \times & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & - (\hat{\omega}_t) \times \end{bmatrix} \]

\[ \frac{d}{dt} \xi_t = A_t \xi_t + w_t \]

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Forward Kinematic Position Measurements

Using forward kinematics, we can measure the position of the contact frame relative to the base (IMU) frame:

\[ Bp_{BC}(\tilde{\alpha}_t) \approx R_t^T(d_t - p_t) + B\dot{J}_{BC}(\tilde{\alpha}_t)w^\alpha_t \]

Written in matrix form:

\[
\begin{bmatrix}
Bp_{BC}(\tilde{\alpha}_t)
\end{bmatrix}
= \begin{bmatrix}
R_t^T & -R_t^T v_t & -R_t^T p_t & -R_t^T d_t
\end{bmatrix}
\begin{bmatrix}
0_{1\times3}
1
0_{1\times3}
0_{1\times3}
\end{bmatrix}
\begin{bmatrix}
0_{3\times1}
0
1
-1
\end{bmatrix}
+ \begin{bmatrix}
B\dot{J}_{BC}(\tilde{\alpha}_t)w^\alpha_t
\end{bmatrix}
\]

Has right invariant observation structure!

Linearized observation matrix is constant!

\[ H_t = \begin{bmatrix} 0 & 0 & -I & 1 \end{bmatrix} \]

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Right Invariant EKF Equations

Propagation:

\[
\frac{d}{dt} \bar{X}_t = f_{u_t}(\bar{X}_t) \\
\frac{d}{dt} P_t = A_t P_t + P_t A_t^T + \bar{Q}_t,
\]

Correction:

\[
\bar{X}_t^+ = \text{Exp}\left( K_t \Pi \left( \bar{X}_t Y_t \right) \right) \bar{X}_t \\
P_t^+ = (I - K_t H_t) P_t
\]

Linearizations are constant!

\[
A_t = \begin{bmatrix}
0 & 0 & 0 & 0 \\
(g) & 0 & 0 & 0 \\
0 & I & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\bar{Q}_t = \text{Ad}_{X_t} \text{Cov}(w_t) \text{Ad}_{X_t}^T
\]

\[
S_t = H_t P_t H_t^T + \bar{N}_t \\
K_t = P_t H_t^T S_t^{-1}
\]

\[
H_t = \begin{bmatrix}
0 & 0 & -I & I
\end{bmatrix}
\]

\[
\bar{N}_t = \bar{R}_t B J_{\text{BC}}^\hat{\beta}(\hat{\alpha}_t) \text{Cov}(w_t^\alpha) \left( B J_{\text{BC}}^\hat{\beta}(\hat{\alpha}_t) \right)^T \bar{R}_t^T
\]

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Observability Analysis

Discrete time state transition matrix:

\[
\Phi = \exp_m(A_t \Delta t) = \begin{bmatrix}
I & 0 & 0 & 0 \\
(g) \times \Delta t & I & 0 & 0 \\
\frac{1}{2} (g) \times \Delta t^2 & I \Delta t & I & 0 \\
0 & 0 & 0 & I
\end{bmatrix}
\]

Observability matrix can be computed as:

\[
\mathcal{O} = \begin{bmatrix}
H \\
H\Phi \\
H\Phi^2 \\
\vdots
\end{bmatrix} = \begin{bmatrix}
0 & 0 & -I & I \\
-\frac{1}{2} (g) \times \Delta t^2 & -I \Delta t & -I & I \\
-2 (g) \times \Delta t^2 & -2I \Delta t^2 & -I & I \\
\vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\]

- Absolute position and yaw are unobservable (drift will occur)
- Remaining states have local stability about any trajectory!
Simulation Results

We ran 100 simulations using the same measurements and noise statistics, while randomly initializing the orientation and velocity estimates.

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Covariance Propagation

Robot walks forward with initial yaw uncertainty

Position uncertainty cannot be captured with a simple covariance ellipse.
Walking with Unknown Initial Yaw

- Robot walking in a straight line with completely uncertain initial yaw.
- Yaw is unobservable along with absolute position

\[ \text{Exp}(\xi_t) = \eta_t \]
Incorporating IMU Bias

Unfortunately, no known way to incorporate IMU bias into the Lie group while maintaining the “group affine” property.

**“Imperfect” Invariant EKF**

State and errors become tuples:

\[(X_t, b_t) \in G \times \mathbb{R}^6\]

\[e_t \triangleq (\bar{X}_t X_t^{-1}, \bar{b}_t - b_t)\]

Invariant EKF | Error-State EKF

New linearized dynamics and noise matrices:

\[
A_t = \begin{bmatrix}
0 & 0 & 0 & 0 \\
(g) \times & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\bar{Q}_t = \begin{bmatrix}
\text{Ad}_{X_t} & 0_{12,6} \\
0_{6,12} & I_6
\end{bmatrix} \text{COV}(w_t) \begin{bmatrix}
\text{Ad}_{X_t} & 0_{12,6} \\
0_{6,12} & I_6
\end{bmatrix}^T
\]

[Barrau 2015]

[Hartley et al. RSS 2018]

[Hartley et al. IJRR 2019]
Experimental Results

We ran the filters 100 times using the same measurements (from a walking experiment) and noise statistics, while randomly initializing the orientation and velocity estimates.
Motion Capture Experiment

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Motion Capture Experiment

Orientation (exponential coordinates)

[Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
Motion Capture Experiment

Velocity (body frame)

$v_x$ (m/s)

$v_y$ (m/s)

$v_z$ (m/s)

[Refs: Hartley et al. RSS 2018]
[Hartley et al. IJRR 2019]
New Torso and Perception System

- Velodyne 32 beam LiDAR (10 Hz)
- Ouster 64 beam LiDAR+IMU (10 Hz)
- Two Intel RealSense depth cameras (30 Hz)
- VectorNav-100 IMU (800 Hz, in pelvis)
- Nvidia Jetson TX2 GPU
- Router, switch, power supply

**Challenges:** calibration, synchronization, data collection
LiDAR Motion Compensation using InEKF

We use the high-frequency odometry from the InEKF to correct for Cassie’s movement within single LiDAR scans.

without motion compensation

with motion compensation
InEKF SLAM with Landmarks

Orientation, velocity and position of IMU in world frame

\[
X_t \triangleq \begin{bmatrix}
R_{WB}(t) & wv_{WB}(t) & wp_{WB}(t) \\
0_{1,3} & 1 & 0 \\
0_{1,3} & 0 & 1 \\
0_{1,3} & 0 & 0 \\
\vdots & \vdots & \vdots \\
0_{1,3} & 0 & 0 \\
0_{1,3} & 0 & 0 \\
\vdots & \vdots & \vdots \\
0_{1,3} & 0 & 0 \\
\end{bmatrix}
\]

Position of contact points in world frame

\[
\begin{bmatrix}
wP_{WC_1}(t) & \cdots & wP_{WC_N}(t) \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0 \\
\end{bmatrix}
\]

Position of landmarks in world frame

\[
\begin{bmatrix}
wP_{WL_1}(t) & \cdots & wP_{WL_N}(t) \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0 \\
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 1 \\
\end{bmatrix}
\]
Additional Types of Invariant Measurements

- Landmark Measurement (right invariant) [Zhang 2017]
- GPS Measurement (left invariant) [Barczyk 2011] [Barrau 2015]
- Magnetometer Measurement (right invariant) [Barczyk 2011] [Barrau 2015]
- Position, Velocity, or Pose Measurement (right or left invariant)

Open source C++ library
(https://github.com/RossHartley/invariant-ekf)

Extendable to many aided-inertial navigation systems (wheeled or flying robots!)
The pose of the robot is estimated from Invariant EKF odometry in the IMU frame.
Extends to Factors Graphs

Unary Forward Kinematic Factor

Binary Contact Factor

Unary Forward Kinematic Factor

[Hartley et al. IROS 2018]
Visual-Inertial-Contact Factor Graph

\[
\begin{align*}
\text{minimize} & \quad \sum_{k} \frac{1}{\Sigma_{0}} || r_0 ||^2 + \sum_{i,j \in \mathcal{K}_k} \frac{1}{\Sigma_{L_{ij}}} || r_{L_{ij}} ||^2 + \sum_{i,j \in \mathcal{K}_k} \frac{1}{\Sigma_{L_{ij}}} || r_{L_{ij}} ||^2 + \sum_{i,j \in \mathcal{K}_k} \frac{1}{\Sigma_{L_{ij}}} || r_{L_{ij}} ||^2 + \sum_{i \in \mathcal{K}_k} \frac{1}{\Sigma_{r_{F_i}}} || r_{F_i} ||^2 + \sum_{i \in \mathcal{K}_k} \frac{1}{\Sigma_{r_{F_i}}} || r_{F_i} ||^2 + \sum_{i,j \in \mathcal{K}_k} \frac{1}{\Sigma_{C_{ij}}} || r_{C_{ij}} ||^2 + \sum_{i,j \in \mathcal{K}_k} \frac{1}{\Sigma_{C_{ij}}} || r_{C_{ij}} ||^2
\end{align*}
\]

Prior

Loop closure

IMU

Visual

Forward Kinematic

Hybrid Contact

[Hartley et al. IROS 2018]
Thank you!
Questions?