VILENS – The Challenge of Visual Navigation on Legged Robots

Dr. Maurice Fallon,
Lecturer / Royal Society Research Fellow

Research by: David Wisth and Dr. Marco Camurri
Dynamic Robot Systems Group, Oxford Robotics Institute
State estimation – target applications

Muddy Tunnel – DARPA SubT (x3)

City of Zurich Sewer

Videos: courtesy of DARPA & ETH
State estimation – the challenge

Dynamic Walking – at 0.5m/sec on Flat Ground
Precise Estimate Not Important

Static Stair Climbing
Precise Estimate due to slow motion

Limitations for Dynamic Control e.g. stair climbing:
• Stronger leg motors. Lighter robot
• More dynamic control algorithms
• Poor elevation mapping due to poor state estimation

State Estimate for Dynamic Gaits has high drift due to:
• Contact Classification Errors. Structural bending
• Rough contact events. **Soft ground**
State estimation – the effect

- Essential for control, motion planning, navigation, etc.
- Traditionally, legged robots rely on kinematic-inertial inputs and suffer from drift.

Estimate from Kinematic-Inertial Estimator (TSIF)
State estimation – the effect

- This drift can cause issues for other systems (e.g. terrain mapping)

Estimate from Kinematic-Inertial Estimator (TSIF)
Goal: To estimate the state of the robot:

- Position and orientation of base (6)
- **Linear and Angular Velocities of base** (6)
- Angular Rate and Acceleration biases (6)
- Joint angles and velocities (N*2)

**For a quadruped:**
- $6*2 + 12*2 + 6$ biases = 36 states

**For a biped:**
- $6*2 + 27*2 + 6$ biases = 72 states
**Kinematic State Estimation - Overview**

**Goal:** To estimate the **state of the robot’s base**
- Position, orientation and velocity
- High frequency (250/400 Hz)
- Low latency (2-3 msec)

**Main Challenges:**
- Despite modelling, behaviour in contact is unknown
- **Latency is unacceptable:** estimate used in control

**Core Approach:** (Extended) Kalman Filter
- IMU-driven process model
- Leg kinematics used to measure **linear velocity**
- **No external sensors (cameras, LIDAR)**
- **Crucial:** contact classification using forces

**Implementations:**
- Anybotics/RSL: TSIF
- Oxford: Pronto

**Diagram:**
- Gyro/accel sensing
- 400Hz IMU Process/Prediction
- 400Hz Kinematic Measurement
- Kinematics / foot sensing
- to controller
Kinematic State Estimation - Examples

Continuous Humanoid Locomotion Enabled by Online Footstep Planning and Stereo Fusion

With Boston Dynamics Atlas Biped
A few cm of drift after 15m travelled
Used in DARPA Robotics Challenge

With IIT HyQ Quadruped
PhD Research of Marco Camurri
Kinematic State Estimation - Challenges

Biped: foot force sensing state model

Quadruped: spurious raw velocities
Kinematic State Estimation - Overview

Goal: To estimate the **state of the robot’s base**
- Position, orientation and velocity
- High frequency (250-400 Hz)
- Low latency (2-3 msec)

Main Challenges:
- Despite modelling, behaviour in contact is unknown
- **Latency is unacceptable**: estimate used in control

Core Approach: (Extended) Kalman Filter
- IMU-driven process model
- Leg kinematics used to measure **linear velocity**
- **Crucial**: contact classification using forces
- **No external sensors (cameras, LIDAR)**

Implementations:
- Anybotics/RSL: TSIF
- Oxford: Pronto

Diagram:
- IMU & kinematics -> TSIF / Pronto (estimator) -> Drifting state estimate
- Drifting state estimate -> Elevation Mapping
- Elevation Mapping -> Controller

Controller

Elevation mapping
Aim to combine Leg Odometry and VIO

Legged Robot State Estimation

• Typically fused with Kalman Filters:
  • TSIF [Bloesch] - RSL / ANYbotics
  • Pronto [Nobili] – MIT / Oxford
• Typically uses high grade sensors.

Visual-Inertial Odometry

• Mature field
• Good results on datasets
• Sliding window optimisation popular (e.g. ROVIO, VINS-Mono)

Bloesch et. al: State Estimation for Legged Robots – Consistent Fusion of Leg Kinematics and IMU. RSS 2013
Nobili et. al: Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots. RSS 2017
Visual Inertial Legged Navigation System - Contributions

- VILENS - First algorithm to tightly fuse vision, IMU, and leg odometry
- Extensive Testing with 250+ m experiments
- Consumer Grade Cameras (Intel RealSense)

Previous Systems:
- legged odometry centred
  - with optional vision (or LIDAR)

Proposed System:
- vision centred
  - with optional leg odometry
Direction of Progress: COTS depth/stereo cameras

ANYbotics ANYmal Version C

Boston Dynamics Spot
VILENS (Visual Inertial Legged Navigation System)

- IMU (400Hz) $\mathcal{I}_{\Delta i}$
- Camera (30Hz) $\mathcal{C}_i$
- Kinematics (400Hz) $\mathcal{Q}_{\Delta i}$

VILENS

Factor Graph Optimization

- State estimate

\[ x_i \triangleq [R_i, p_i, v_i, b_i] \]

Control, Mapping, Navigation, ...

UNIVERSITY OF OXFORD
VILENS (Visual Inertial Legged Navigation System)

- IMU (400Hz)
- Camera (30Hz)
- Kinematics (400Hz)

Control, Mapping, Navigation, ...

\[ \chi^* = \arg \min_{\chi} \| r_0 \|^2 \Sigma_0 + \sum_{i \in K_k} \| r_{I \Delta i} \|^2 \Sigma_{I \Delta i} + \sum_{j \in M} \| r_{m_{j,0}} \|^2 \Sigma_{m_{j,0}} + \sum_{i \in K_k} \sum_{j \in M_i} \| r_{m_j} \|^2 \Sigma_{m_j} + \sum_{i \in K_k} \| r_{Q \Delta i} \|^2 \Sigma_{Q \Delta i} \]
Factor Definitions

**iSAM2 – using GTSAM:**
- Sliding window batch optimization

**Prior Factors:**
- Set initial conditions

\[
r_0(x_0, Z) = \begin{pmatrix}
    \Phi(T_0^{-1} T_{p_0}) \\
    v_0 - v_{p_0} \\
    b_0^a - b_{p_0}^a \\
    b_0^3 - b_{p_0}^\omega
\end{pmatrix}
\]

**IMU Factors (Forster et al):**
- Difference between IMU preintegration & estimate, w/ biases.

\[
r_{L\Delta_i} = [r_{\Delta R_{\Delta_i}}, r_{\Delta p_{\Delta_i}}, r_{\Delta v_{\Delta_i}}, r_{\Delta b_{\Delta_i}}]
\]

**Leg Odometry Factors:**
- Assuming the contact points are fixed, estimate the relative motion.
- Formulate output of existing kinematic-inertial TSIF estimator (Bloesch et al.) as a relative pose constraint.

\[
r_{Q_{\Delta_i}} = \Phi \left( (T_{i-1}^{-1} T_i)^{-1} \tilde{T}_{i-1}^{-1} \tilde{T}_i \right)
\]

Feature Tracking Front-End:
- Track feature through successive frames (KLT feature tracker).

Vision Factors:
- Estimate 3D location of landmarks.
- Minimise reprojection error between estimate and measured:

\[
\mathbf{r}_{m_j} = \left( \begin{array}{c}
\pi_u(\mathbf{R}_i, \mathbf{p}_i, m_j) - u_{i,j} \\
\pi_v(\mathbf{R}_i, \mathbf{p}_i, m_j) - v_{i,j}
\end{array} \right)
\]
- Add prior to help under-constrained landmarks:

\[
\mathbf{r}_{m_j,0} = m_j - m_{j,0}
\]
• Standard visual-inertial odometry dataset.
• Qualitative results, demonstrating the system can function as a stand-alone VINS system.
• Comparable performance to state of the art VINS systems.
Experimental Setup - ANYmal

Oil Rig Training Site: Realistic Industrial Mock-up
250m of continuous walking. Brightness variation. Climbing, Trotting
Experimental Setup – Ground Truth

Leica TS-16 - Ground Truth Tracking
Robust Legged Robot State Estimation Using Factor Graph Optimization

David Wissh, Marco Camurri, Maurice Fallon

Oxford Robotics Institute – University of Oxford
Analysis + Discussion

• We outperform the baseline kinematic-inertial estimator (TSIF)
  • 55% in RPE and 76% in ATE.
• Our algorithm operates even when off-the-shelf VIO fails.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RPE $\mu(\sigma)$ [m]</th>
<th>Yaw Error $\mu(\sigma)$ [deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TSIF [3]</td>
<td>VILENS</td>
</tr>
<tr>
<td>Keble 1</td>
<td>0.53 (0.21)</td>
<td>0.30 (0.12)</td>
</tr>
<tr>
<td>Keble 2</td>
<td>0.51 (0.10)</td>
<td>0.23 (0.10)</td>
</tr>
<tr>
<td>Keble 3</td>
<td>0.67 (0.10)</td>
<td>0.52 (0.15)</td>
</tr>
<tr>
<td>Keble 4</td>
<td>0.47 (0.11)</td>
<td>0.40 (0.10)</td>
</tr>
<tr>
<td>Oil Rig</td>
<td>0.44 (0.37)</td>
<td>0.41 (0.18)</td>
</tr>
<tr>
<td></td>
<td>TSIF [3]</td>
<td>VILENS</td>
</tr>
<tr>
<td></td>
<td>6.64 (2.23)</td>
<td>0.99 (0.80)</td>
</tr>
<tr>
<td></td>
<td>5.72 (0.94)</td>
<td>1.47 (1.07)</td>
</tr>
<tr>
<td></td>
<td>6.68 (0.80)</td>
<td>3.86 (1.90)</td>
</tr>
<tr>
<td></td>
<td>3.32 (1.15)</td>
<td>1.13 (1.46)</td>
</tr>
<tr>
<td></td>
<td>4.89 (3.38)</td>
<td>3.68 (4.10)</td>
</tr>
</tbody>
</table>
Leg Odometry Biases

TSIF: bad yaw bias estimation
VILENS: good yaw bias estimation

TSIF: drift in height. Poorly observed
Visual Inertial Legged Navigation (VILENS) - 2020

Previous Factor Graph (IROS 2019)

Current Factor Graph (Under Review)

\[
\chi^*_k = \arg \min_{\chi_k} \| r_0 \|_2 \sum_{i=0}^{N} + \sum_{l \in K_k} \| r_{\mathcal{L}_{ij}} \|_2 + \sum_{i \in K_k} \| r_{\mathcal{V}_{ij}} \|_2 + \sum_{i \in K_k} \| r_{b_{ij}} \|_2 + \sum_{i \in K_k} \sum_{\ell \in M_i} \| r_{i,m_{\ell}} \|_2
\]

\[
\| r_{b_{ij}} \|_2 \triangleq \| b_{i}^q - b_{i-1}^q \|_2 \| b_{i}^a - b_{i-1}^a \|_2 \| b_{i}^w - b_{i-1}^w \|_2 \| b_{i}^v - b_{i-1}^v \|_2
\]

Gyro ang. velocity
Accelerometer
Leg ang. velocity
Leg linear velocity
Preintegrated Velocity Bias Estimation to Overcome Contact Nonlinearities in Legged Robot Odometry

David Wisth, Marco Camurri, Maurice Fallon

Oxford Robotics Institute - University of Oxford

Factor Graph Optimization

Visual features

Leg position

Inertial sensing

Preintegrated Velocity Bias Estimation to Overcome Contact Nonlinearities in Legged Robot Odometry

Tracker (Ground Truth)

Intel RealSense Depth Camera
D435i Active Depth Camera + IMU
Integration into real-time system

**Default System**
- Robot
- imu & kinematics
- TSIF / Pronto (estimator)
- Drifting state estimate
- Elevation Mapping
- Controller

**Proposed System**
- Robot
- imu & kinematics
- Camera images
- VILENS (visual estimator)
- Bias feedback – not implemented yet
- Better state estimate
- Elevation Mapping
- Better elevation map
- Controller
Local Terrain Map Improvement

Using vision in estimation improves terrain map

VILENS running live on ANYmal
Local Terrain Map Improvement

TSIF map is very inaccurate

VILENS accurate reconstruction
Local Terrain Map Improvement - Live

Work with RSL (ETH)
Implementation

• iSAM2 incremental optimizer (part of GTSAM library).
• Zero velocity states based on average feature movement.
• Works reliably with D435 on the robot at 15Hz
  • Exploring using RealSense T265
  • Considering mono-fisheye

• Multiple Sensor and Processing Message Threads
• Tested on three different copies of ANYmal:
  • Oxford
  • RSL (ETH)
  • ANYbotics
  • Contributing to ANYmal’s SLAM in DARPA SubT Cerberus
Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure

Milad Ramezani, Georgi Tinchev, Egor Iuganov and Maurice Fallon

Oxford Robotics Institute, University of Oxford

LIDAR Pose-Graph SLAM with Deep Learning Loop Closure
Thank You

For further information:
David Wisth
davidw@robots.ox.ac.uk

Data with ground truth:
ori.ox.ac.uk/vilens

Thanks to:
RSL (ETH) & ANYbotics