Legged State Estimation

Marco Camurri – OXFORD ROBOTICS INSTITUTE

UNIVERSITY OF OXFORD

MEMMO Winter School
Idiap Research Institute – Martigny, Switzerland
Thursday 31st , 2019

Outline

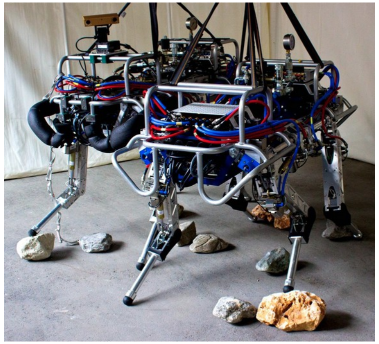
- Intro Proprioceptive State Estimation
- Inertial-Legged Estimation with Pronto
- Pronto Applications
- Other methods

About me

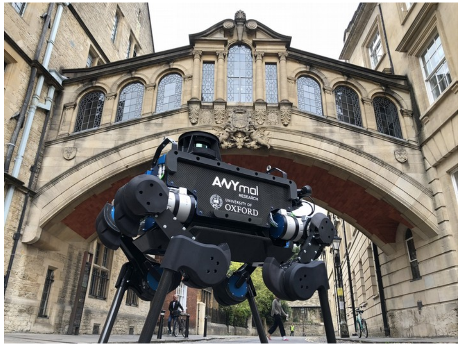
ORI autonomous systems

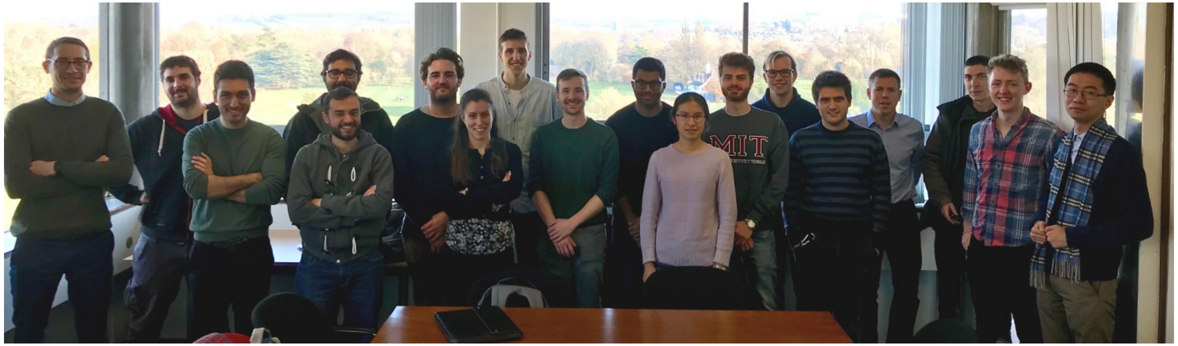
UNIVERSITY OF OXFORD

2014 – 2018: PhD/Postdoc at IIT (Genoa, Italy)



2018: moved to ORI for Postdoc





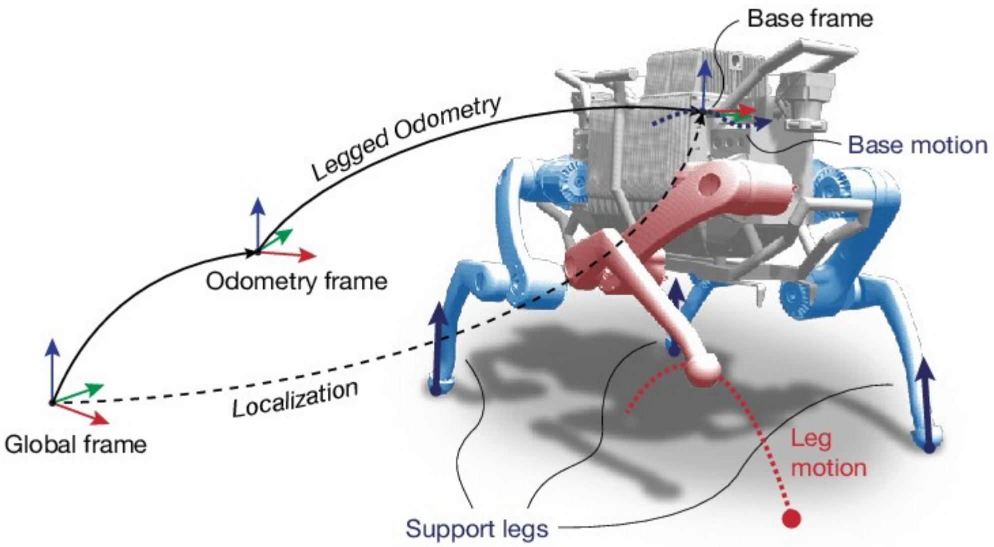
Outline

- **Intro Proprioceptive State Estimation**
- Inertial-Legged Estimation with Pronto
- Pronto Applications
- Other methods

What is state estimation?

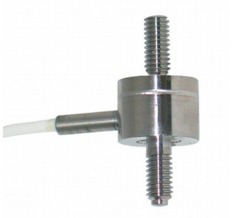
Estimate the **position** and **velocity** of a floating base legged robot from sensor measurements

- High frequency (~0.2-1 kHz)
  → **control**
- Low frequency (~1-90 Hz)
  → **localization, mapping, path planning**



Proprio- vs. Extero-ceptive

Proprioceptive state estimation → properties of the robot

NOTE: images not in scale

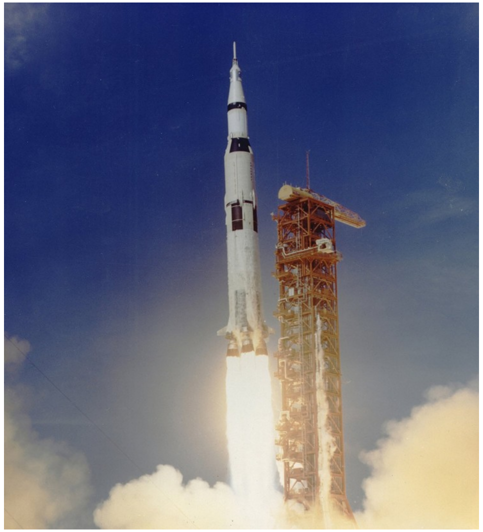
Exteroceptive state estimation → properties of the environment


NOTE: images not in scale

Normally, **both are used** either in a **loosely or tightly coupled** system

Strapdown Inertial Navigation

Strapdown = IMU on the chassis



$$\Theta_k = \Theta_{k-1} \exp(\boldsymbol{\omega}_k^{\wedge} \Delta t)$$

$$\Theta_k = \Theta_{k-1} \left(I_3 + \frac{\sin\sigma}{\sigma} \boldsymbol{\omega}_k^{\wedge} + \frac{1-\cos\sigma}{\sigma^2} (\boldsymbol{\omega}_k^{\wedge})^2 \right)$$

$$\ddot{\mathbf{x}}_k = \Theta_k \mathbf{a}_k - {}_w\mathbf{g}$$

$$\dot{\mathbf{x}}_k = \dot{\mathbf{x}}_{k-1} + \Delta t \ddot{\mathbf{x}}_k$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta t \dot{\mathbf{x}}_k$$

$$\boldsymbol{\omega}^{\wedge}(t) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & \omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

What is Leg Odometry?

Estimation of incremental motion of a legged robot through kinematics (and dynamics) assuming no slip

- **How to detect contacts**
  - Measured force (force/torque sensors)
  - Estimated force (from dynamics)
  - other
- **Meas. model/State Definition**
  - Foot positions in state
  - Velocity updates
- **How to fuse measurements**
  - No fusion
  - EKF, UKF, xKF
  - Optimization (QP, TSIF, F. Graph)
- **Leg fusion**
  - Multiple legs
  - One leg

What is Leg Odometry?

- Quasi-Static motion
- Six legs → overconstr.
- One Swing leg
- No fusion with IMU

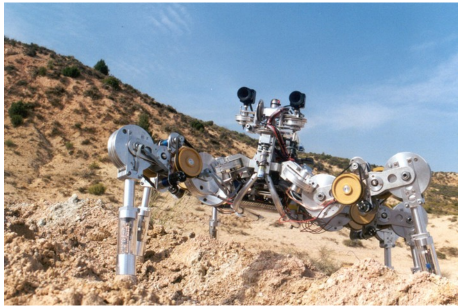
$$\min \sum_j w_j \mathbf{e}_j^{\mathrm{T}} \mathbf{e}_j$$

$$\mathbf{e}_j = {}^W\mathbf{p}_j - {}^W R_B \, {}^B\mathbf{p}_j + {}^W\mathbf{t}$$

Early State Estimation (90s/10s)

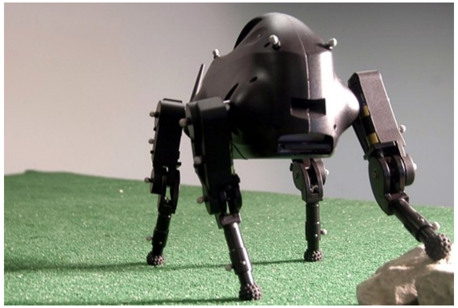
- Six legs (tripod gait)
- Flat Ground
- Mild GPS/VO integr.
- MoCap aid

Crawler - DLR



Lauron III - FIK

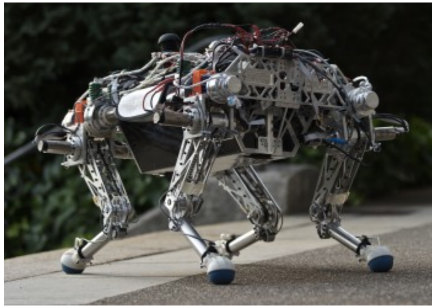
RHex - BDI



LittleDog - BDI

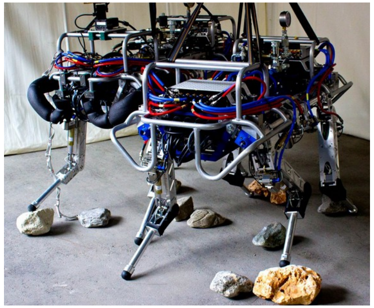
Modern State Estimation ('10-now)

LS3 - BDI



StarlETH - ETHZ

BigDog - BDI



HyQ - IIT

- Inertial-Kinematics
- Kalman Filter-based
- No assumptions on leg configuration or gait
- (w/o) contact sensors
- Exteroceptive fusion

State Estimation on Humanoids

- Additional joint (ankle)
- Flat foot (foot orientation, torque)
- 4 force/torque sensors at the foot/spring at ankle
- Non-negligible flexibilities

Proprioceptive State Estimation

- **Extended/Unscented Kalman Filter** → Inertial Process Model + Leg Odometry
- **Dynamic State Estimation via Quadratic Programming** → minimize model/process error
- **Two State Implicit Filter** → no prediction phase

Unobservable States

- Absolute position and yaw are unobservable
- Roll and pitch are observable → gravity
- Twist is measured/observable → IMU

Position and yaw drift without exteroceptive updates

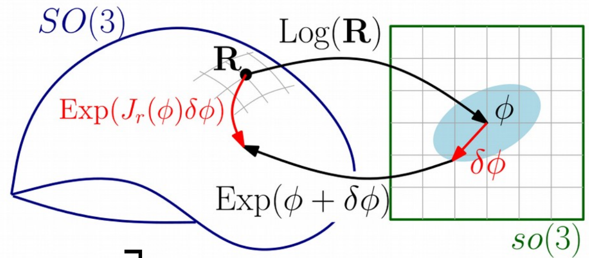
We can limit the drift rate …

"State Estimation for Legged Robots – Consistent Fusion of Leg Kinematics and IMU"

M. Bloesch, M. Hutter, M.A. Hoepflinger, S. Leutenegger, C. Gehring, C.D. Remy, R. Siegwart – RSS 2012

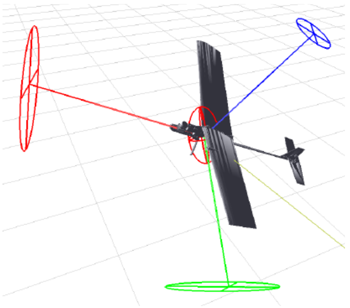
Inertial Process Model

- How to parametrize orientation?
  - Rotation Matrix / Quaternion for state, Euler for error

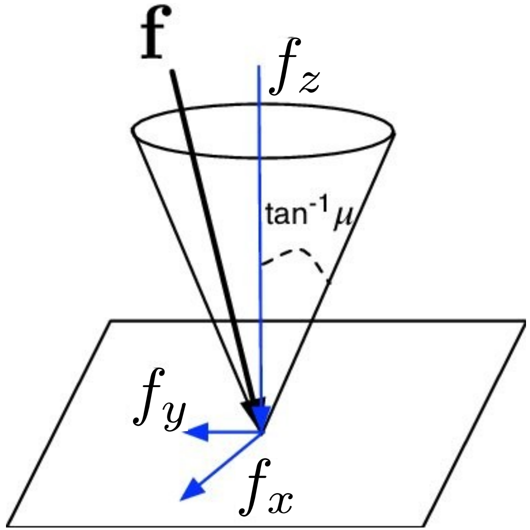


$$\boldsymbol{u}_k = \begin{bmatrix} \tilde{\boldsymbol{\omega}}_k \\ \ddot{\tilde{\mathbf{x}}}_k \end{bmatrix} = \begin{bmatrix} R_i^b({}_i\boldsymbol{\omega}_i - \mathbf{b}^+_{\omega,k-1}) \\ R_i^b({}_i\ddot{\mathbf{x}}_i - \mathbf{b}^+_{a,k-1}) \end{bmatrix}$$

$$\boldsymbol{x}_k^- = \begin{bmatrix} \mathbf{x}_k^- \\ \dot{\mathbf{x}}_k^- \\ \Theta_k^- \\ {}_a\mathbf{b}_k^- \\ {}_\omega\mathbf{b}_k^- \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{k-1}^+ \\ \dot{\mathbf{x}}_{k-1}^+ \\ \mathbf{0} \\ {}_a\mathbf{b}_{k-1}^+ \\ {}_\omega\mathbf{b}_{k-1}^+ \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{x}}_{k-1}^+ \Delta t \\ (-\tilde{\boldsymbol{\omega}}_k \times \dot{\mathbf{x}}_{k-1}^+ + (\Theta_{k-1}^+)^{-1}\, {}_w\mathbf{g} + \ddot{\tilde{\mathbf{x}}}_k)\Delta t \\ \Theta_{k-1}^+ \exp(\boldsymbol{\omega}_k^\wedge \Delta t) \\ \dot{\mathbf{b}}^-_{a,k-1}\Delta t \\ \dot{\mathbf{b}}^-_{\omega,k-1}\Delta t \end{bmatrix}$$



Contact Assumptions



No contact sensors

$$\mathbf{f} = -(J_{cq}^T(\mathbf{q}))^\dagger \left(\boldsymbol{\tau} - \mathbf{h}_q - F^{\mathrm{T}}{}_b\ddot{\mathbf{x}}_b\right)$$

No slippage

$$\frac{\sqrt{f_x^2 + f_y^2}}{f_z} \leq \mu$$

No rotation (flat foot)

$$\begin{bmatrix} -\tau_y / f_z \\ \tau_x / f_z \end{bmatrix} \leq \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

$$|\tau_z| \leq \mu_z f_z$$

Leg Odometry (velocity)

- Base velocity from one leg

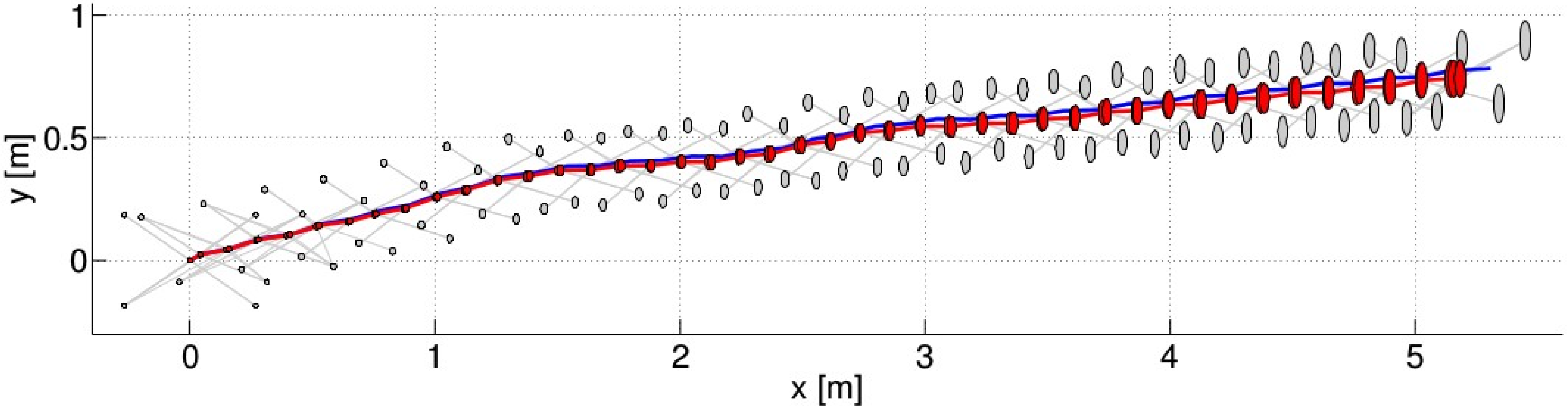
$$\boldsymbol{x} = [\mathbf{x},\ \dot{\mathbf{x}},\ \boldsymbol{\theta},\ \mathbf{b}_\omega,\ \mathbf{b}_a]$$

$$_b\dot{\mathbf{x}}_b = -_b\dot{\mathbf{x}}_f - {}_b\boldsymbol{\omega}_b \times {}_b\mathbf{x}_f$$

Leg Odometry (position)

Include the contact points in the state:

$$\boldsymbol{x} = [\mathbf{x},\ \dot{\mathbf{x}},\ \boldsymbol{\theta},\ \mathbf{p}_1,\ \cdots \mathbf{p}_4,\ \mathbf{b}_\omega,\ \mathbf{b}_a]$$



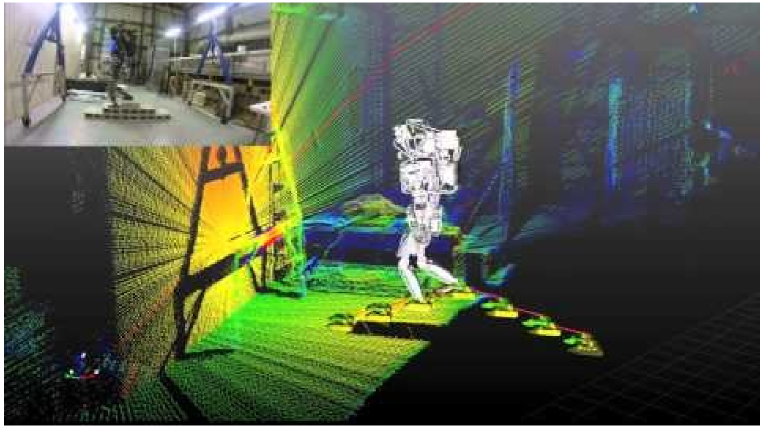
"State Estimation for Legged Robots – Consistent Fusion of Leg Kinematics and IMU"
M. Bloesch, M. Hutter, M.A. Hoepflinger, S. Leutenegger, C. Gehring, C.D. Remy, R. Siegwart – RSS 2012

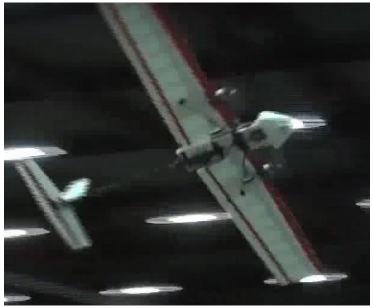
Outline

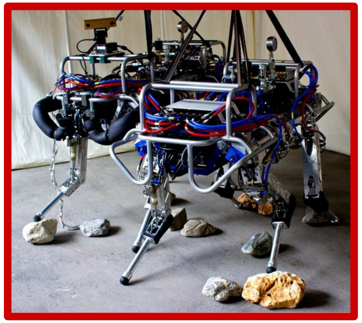
- Intro Proprioceptive State Estimation
- **Inertial-Legged Estimation with Pronto**
- Pronto Applications
- Other methods

Pronto

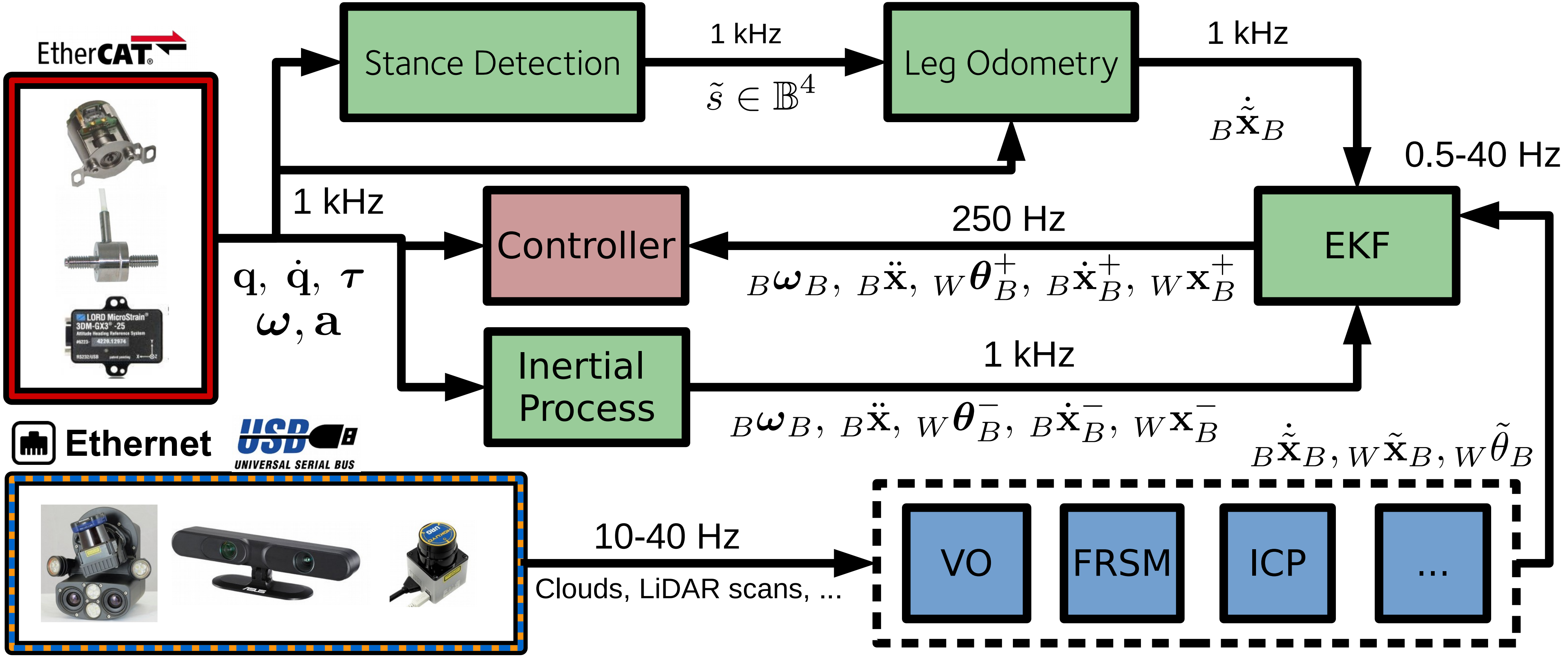
- Extended Kalman Filter
- History of Measurements
- Modular
- Multi-platform





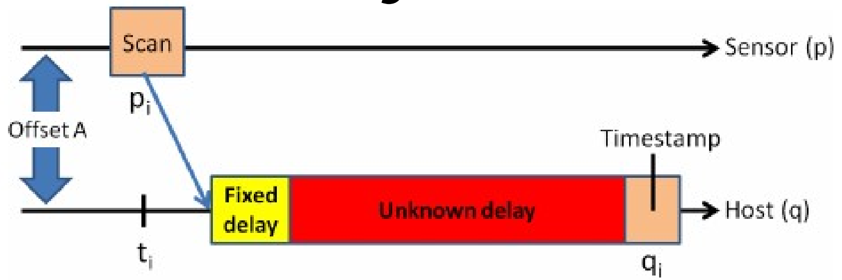


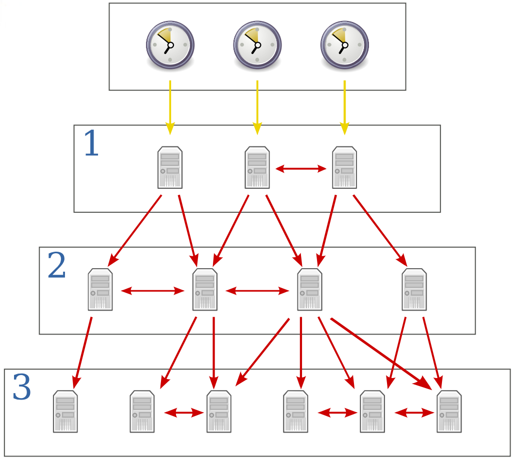
State Estimation Scheme



Time Synchronization

- EtherCAT
- FPGA
- NTP
- Passive Synchronization



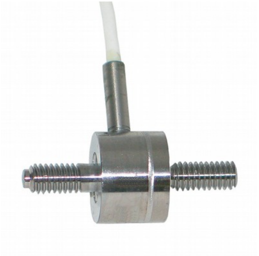


A passive solution to the sensor synchronization problem
E. Olson – IROS 2010

Challenges in Leg Odometry

- Kinematics (joint position, joint velocity)
- IMU (angular velocity, acceleration)

$$ {}_b\dot{\mathbf{x}}_b = -{}_b\dot{\mathbf{x}}_f - {}_b\boldsymbol{\omega}_b \times {}_b\mathbf{x}_f $$



1. Probability of Reliable Contact
2. Measurement Update (from multiple legs)
3. Covariance estimation (for the EKF)

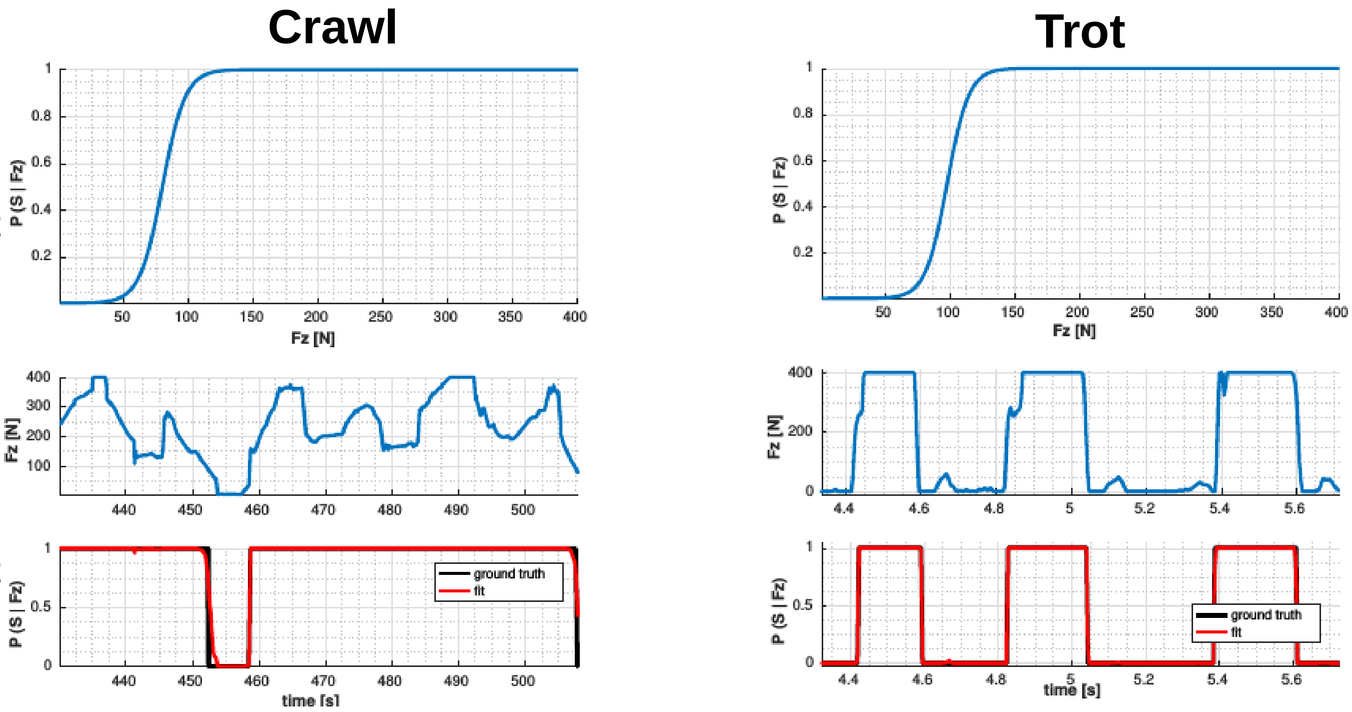
Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots
M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D.G. Caldwell, C. Semini
IEEE Robotics and Automation Letters (RA-L), 2017

Which legs are in reliable contact?

Prob. Contact

Normal Force

Contact state



- No foot sensors → estimate GRF from joint efforts
- What threshold? → minimize velocity error

Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots
M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D.G. Caldwell, C. Semini
IEEE Robotics and Automation Letters (RA-L), 2017

How to fuse legs in reliable contact?

Probability of contact as weighting criteria:

$$\dot{\mathbf{x}}_k = \frac{\sum_{i \in C} \mathcal{P}_k(s_l = 1 | \mathbf{f}_l^k)\dot{\mathbf{x}}_{k,l}}{\sum_{i \in C} \mathcal{P}_k(s_l = 1 | \mathbf{f}_l^k)}$$

$$\mathcal{P}_k(s_l = 1 | \mathbf{f}_l^k) = \frac{1}{1 + \exp(-\beta f_{z,l}^k - \beta_0)}$$

Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots
M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D.G. Caldwell, C. Semini
IEEE Robotics and Automation Letters (RA-L), 2017

How to calculate covariance?

Inter-Leg covariance

$$\Lambda(\sigma_{e,x}, \sigma_{e,y}, \sigma_{e,z})^2 \approx \frac{1}{\dim(C)} \sum_{l \in C} (\dot{\mathbf{x}}_{k,l} - \dot{\mathbf{x}}_k)(\dot{\mathbf{x}}_{k,l} - \dot{\mathbf{x}}_k)^{\mathrm{T}}$$

Impact

$$\Delta_{k,f} = |\Delta \bar{f}_z^k| = \frac{1}{\dim(C)} \sum_{l \in C} |f_{z,l}^k - f_{z,l}^{k-1}|$$

Final estimate

$$\Sigma_v = \Sigma_0 + \left[\frac{1}{2} \Lambda(\sigma_{e,x}, \sigma_{e,y}, \sigma_{e,z}) + I_3 \frac{\Delta_{k,f}}{2\alpha} \right]^2$$

Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots
M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D.G. Caldwell, C. Semini – RA-L 2017

How to calculate covariance?

$$\Sigma_v = \Sigma_0 + \left[\underbrace{\frac{1}{2}\Lambda(\sigma_{e,x}, \sigma_{e,y}, \sigma_{e,z})}_{\text{Leg agreement}} + \underbrace{I_3 \frac{\Delta_{k,f}}{2\alpha}}_{\text{Impact}} \right]^2$$

$$|\Delta \bar{f}_z^k| = \frac{1}{\dim(C)} \sum_{i \in C} |f_{z,l}^k - f_{z,l}^{k-1}|$$

Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots
M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D.G. Caldwell, and C. Semini
IEEE Robotics and Automation Letters, 2017

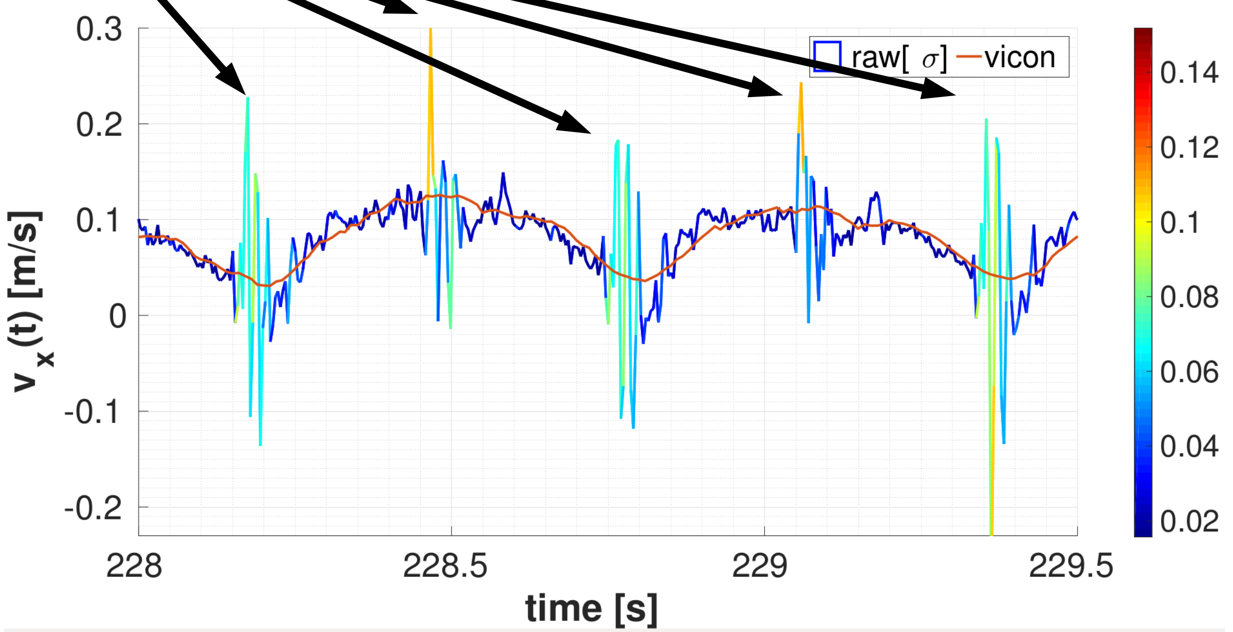
How to calculate covariance?

Covariance boost

Include force discontinuity in the covariance

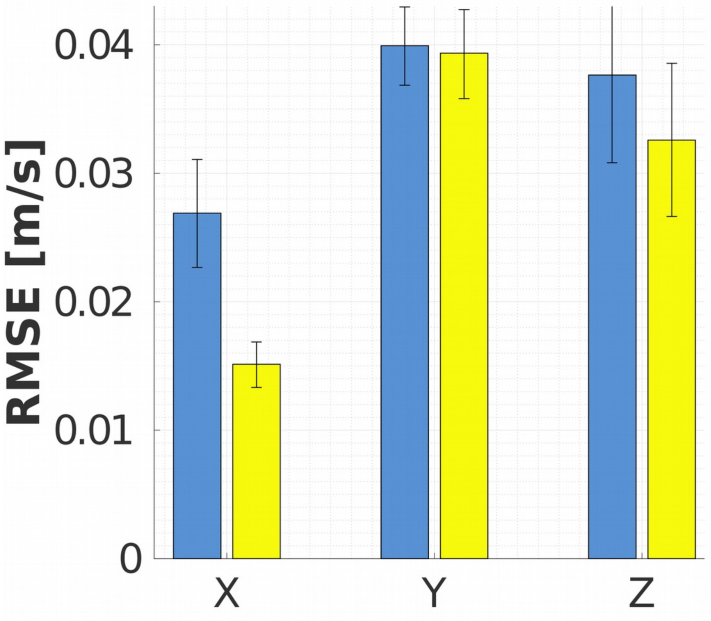
Dark blue
=
low uncertainty

Azure/orange
=
high uncertainty



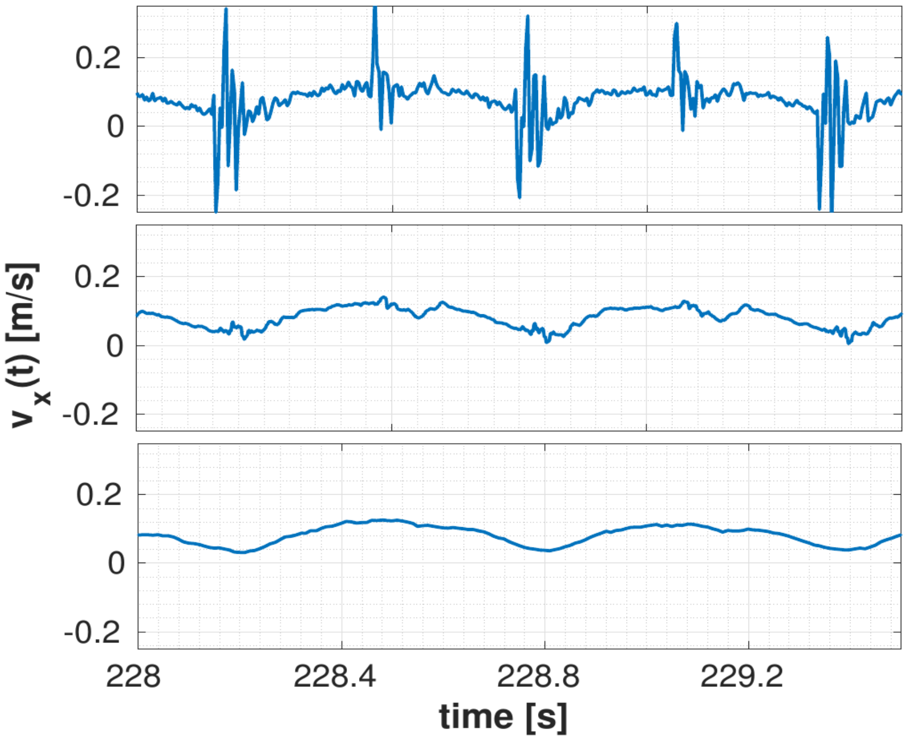
Experimental Results: Velocity

Fixed threshold vs. our approach

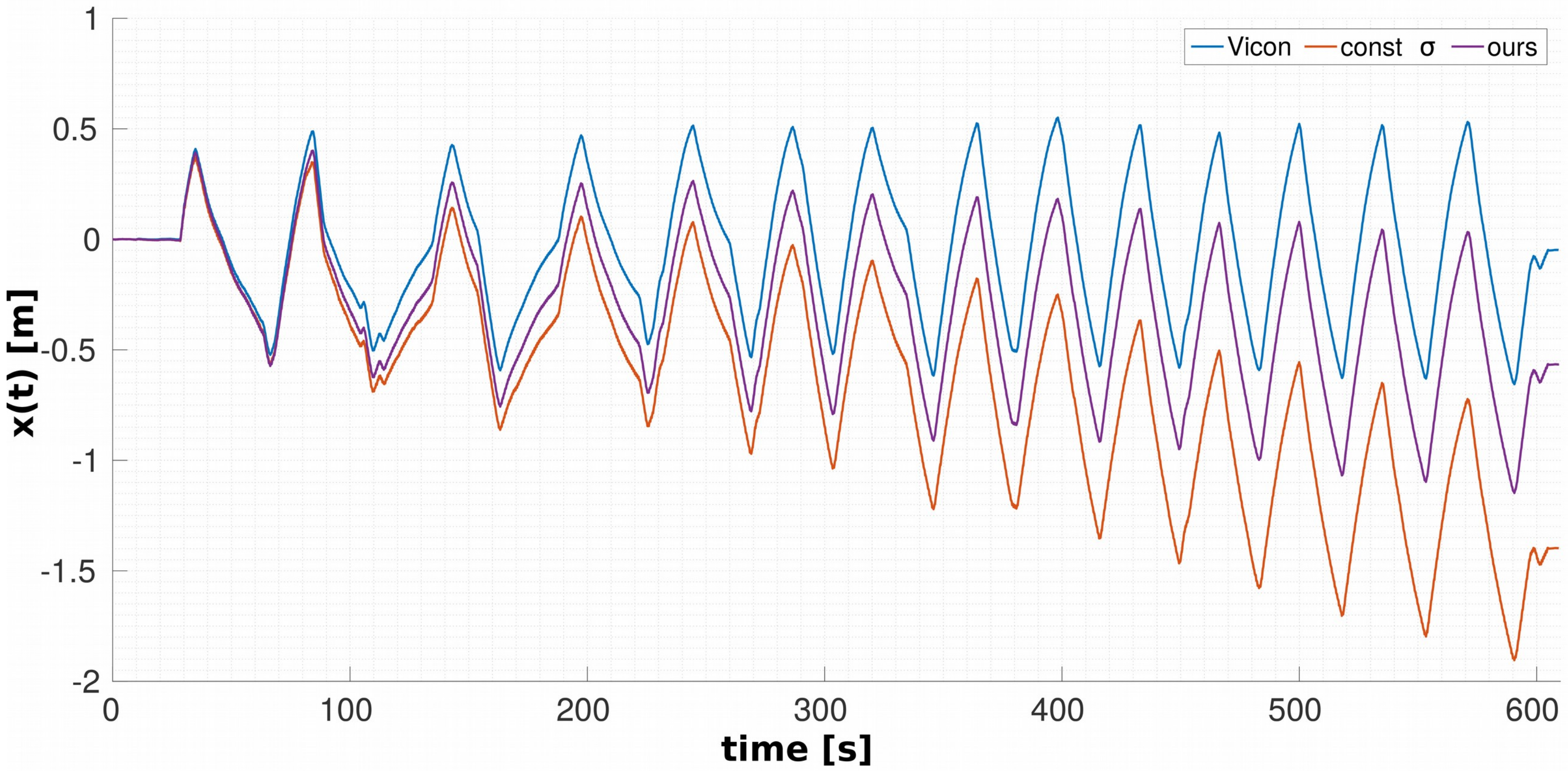


>1h logs, multiple gaits

Velocity forward direction raw vs. filtered vs. ground truth



Experimental Results: Position



Outline

- Intro Proprioceptive State Estimation
- Inertial-Legged Estimation with Pronto
- **Pronto Applications**
- Other methods

Applications

Statically Stable Crawl with height/step reflex

“Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality”

M. Focchi, R. Orsolino, *M. Camurri*, V. Barasuol, C. Mastalli, D.G. Caldwell, C. Semini

To appear in “Springer Tracts in Advanced Robotics”
preprint:
arxiv.org/abs/1805.10238

Applications

Reactive Trotting + CNNs on local height maps

“Fast and Continuous Foothold Adaptation for Dynamic Locomotion through Convolutional Neural Networks”

O. Villarreal, V. Barasuol, <u>*M. Camurri*</u>, M. Focchi, L. Franceschi, M. Pontil, D.G. Caldwell, C. Semini

Accepted to RA-L / ICRA 19, preprint: arxiv.org/abs/1809.09759

Outline

- Intro Proprioceptive State Estimation
- Inertial-Legged Estimation with Pronto
- Pronto Applications
- **Other methods**

Proprioceptive State Estimation

- **Extended/Unscented Kalman Filter →** Inertial Process Model + Leg Odometry
- **Dynamic State Estimation via Quadratic Programming →** minimize model/process error
- **Two State Implicit Filter →** no prediction phase

QP Dynamic State Estimation

QP problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} G \boldsymbol{x} + g^T \boldsymbol{x}$$

$$\text{subject to } C_e \boldsymbol{x} + c_e = 0$$

$$C_i \boldsymbol{x} + c_i \geq 0$$

Cost function $\|A\boldsymbol{x} - b\|^2$

Opt. variables

$$\boldsymbol{x} = \begin{bmatrix} \dot{\mathbf{q}}^{k+1} & \mathbf{w}_k & \boldsymbol{\tau}_k & \mathbf{f}_k \end{bmatrix}^{\mathrm{T}}$$

Model error

$$\mathbf{w} = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) - S\boldsymbol{\tau} - J^{\mathrm{T}}(\mathbf{q})\mathbf{f}$$

$$A = [0 \quad I \quad 0 \quad 0] \qquad \mathbf{b} = \mathbf{0}$$

Measurement error:

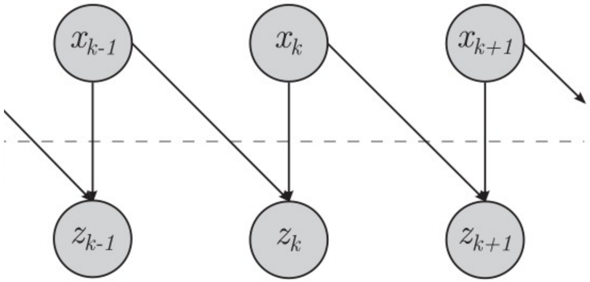
$$A = \text{selector} \qquad \mathbf{b} = \text{measures}$$

Dynamic state estimation using quadratic programming
X. Xinjilefu, S. Feng, and C. Atkeson – IROS 2014

Proprioceptive State Estimation

- **Extended/Unscented Kalman Filter** → Inertial Process Model + Leg Odometry
- **Dynamic State Estimation via Quadratic Programming** → minimize model/process error
- **Two State Implicit Filter** → no prediction phase

Two State Implicit Filter

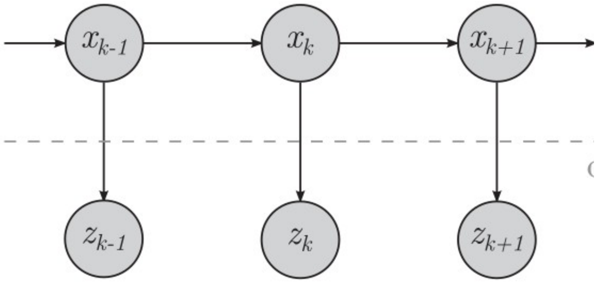


TSIF

$$\left\| \boldsymbol{x}_k^{k-1} - \boldsymbol{x}_{k-1}^{k-1} \right\|_{P_{k-1}}^2 + \left\| \mathbf{b}_k + A\boldsymbol{x}_{k-1}^k + B\boldsymbol{x}_k^k \right\|^2$$

- Current and previous state (two-state) from measurement
- Model can be integrated as "measurement"
- Estimate as residual minimization

The Two-State Implicit Filter Recursive Estimation for Mobile Robots
M. Bloesch, M. Burri, H. Sommer, R. Siegwart, M. Hutter
Robotics and Automation Letters 2018



EKF

$$\text{p.s.} + \left\| F\boldsymbol{x}_{k-1}^k - \boldsymbol{x}_k^k \right\|^2 + \left\| H\boldsymbol{x}_k^k - \boldsymbol{z}_k \right\|^2$$

- Prediction from model
- Update from measurement
- Estimate as "weighted" mean between prediction and update

Any questions?

`mcamurri@robots.ox.ac.uk`