MIT Atlas robot state-estimate

Dehann Fourie

January 14, 2014

1 Introduction

Document to describe the state-estimation process for the Atlas humanoid robot. The proposed approach is an extension of state-of-the art INS/GPS integration techniques. The inertial navigation community would call this a feedback, loosely coupled, complimentary IMU and leg-odometry architecture. This architecture replaces conventional GPS based position measurements with leg kinematics based position estimates.

2 Overview

- This approach favors high frequency inertial information and long term stability of kinematics driven leg-odometry;
- The state-estimate process can be seen as its own closed-loop control system, where errors in an almost pure inertial solution are driven to zero using a EKF based observer;
- Data fusion is achieved by comparing the position (or velocity) estimates of the inertial solution with that of leg-kinematic solution and/or visual-odometry;
- Standard INS error state equations have 15 states: Platform misalignment ρ, gyro biases $\mathbf{b}_g$, local frame velocity error $\delta\mathbf{V}$, local frame position error $\delta\mathbf{P}$ and accelerometer biases $\mathbf{b}_a$.
- The IMU aboard Atlas is classified as tactical grade. I expect $1-5\ ^\circ/hr$ $(1\ \sigma)$ raw performance, which may be improved by as much as one order in the proposed INS/leg-kinematic architecture - providing a preliminary guesstimated North alignment accuracy of $\frac{0.5\ [^\circ/hr]}{15\ [^\circ/hr]\ \cos(latitude)} \times 180\ [^\circ] \approx 8\ ^\circ$, $(1\ \sigma)$. This seems consistent with comments prior to DRC trails. More on this in section 2.1.

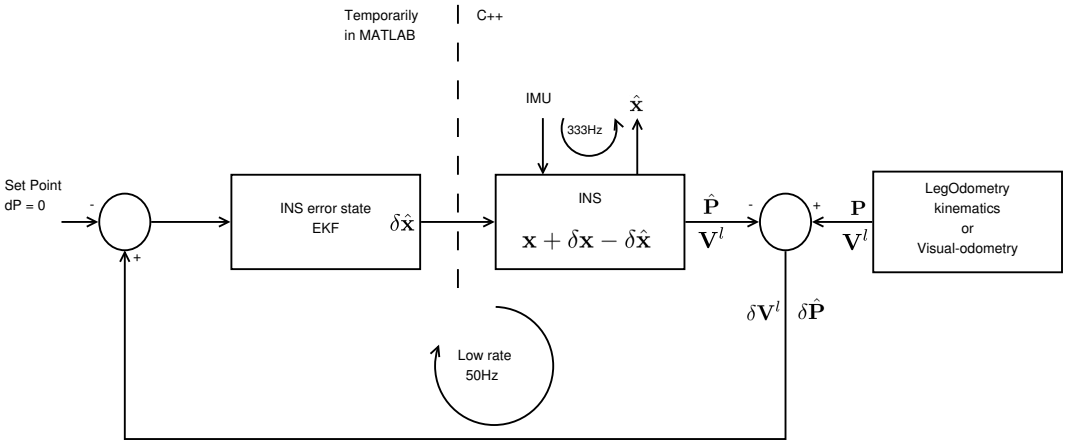


Figure 1: Conceptual overview of state-estimator process

2.1 Selected Notes

- The pure inertial solution is slaved to the long term dynamics of the leg odometry solution – that is the INS will follow the average leg odometry solution over a 2-3min time horizon, below $\frac{1}{2\ mins} \approx 0.008\ Hz$.
- The EKF is to be tuned such that the state estimate in the spectrum between 150 Hz and 0.008 Hz is provided by the almost pure inertial solution;
- The almost pure INS incurs a computational latency of approximately 15 μs, compounded by BDi Atlas + fiber + LCM communication delays. This is the minimum latency method for separated state-estimate and control architectures.
- The pure INS can easily be moved on-board the Atlas vehicle, with an off-vehicle error state estimate, feeding corrections to the inertial based solution.
- This architecture easily supports leg-kinematic based yaw slaving – and will be useful during manipulation tasks. Yaw updates can be added as a measurement to the EKF while Atlas is standing. I propose adding a 16^{th} large heading, Psi-model [1], wander-azimuth alignment state [2] to the EKF to separate the user from true north and local heading issues. ;
- Local frame velocity estimates from visual odometry can be added as measurement to the EKF, as further aiding source to the INS error state – given larger measurement variance to accommodate inaccurate back joint angle measurements.
- Regarding joint backlash – the intention of this approach is to trust the on-line, or dynamically, calibrated inertial solution for the longest possible time span; under the premise that joint encoder errors average to zero on a multi-step basis. The duration of trust on almost pure inertial is directly proportional to the error dynamics of the IMU, which in turn is directly proportional to the cost of the IMU. This is the same 2 $mins$ mentioned above.
- While I do not believe the DRC competition allows scope for experimentation with robust estimation techniques, a growth path for handling joint backlash could be to add robust terms to the INS leg-odometry position measurement model. Prof. J. How could be a good contact point for further investigation of robust terms.

3 Mechanization Equations

3.1 Inertial Navigation

Standard inertial propagation equations are used. Even though there are many structural vibrations on the Atlas robot, we will ignore coning and sculling compensation [3] for the first pass. This is based on [4, 5, 3]

Orientation from local to body is tracked by a quaternion (scalar vector), ${}^b_l\mathring{\mathbf{q}}$, and is propagated with the closed-form, zeroth order solution to the ODE:

$$ {}^b_l\mathring{\mathbf{q}}_{k+1} = \begin{bmatrix} \cos\left(\frac{|\hat{\omega}|}{2}\Delta t\right) \\ \frac{\hat{\omega}}{|\hat{\omega}|}\sin\left(\frac{|\hat{\omega}|}{2}\Delta t\right) \end{bmatrix} \otimes {}^b_l\mathring{\mathbf{q}}_k \tag{1} $$

with quaternion product $\otimes$. We use Taylor expansion approximation when $|\hat{\omega}|$ approaches zero – that is $|\hat{\omega}| < 10^{-7}$:

$$ \frac{\hat{\omega}}{|\hat{\omega}|}\sin\left(\frac{|\hat{\omega}|}{2}\Delta t\right) \approx \left[\Delta t\left(\frac{1}{2} - \frac{|\hat{\omega}|^2\Delta t^2}{48} + \dots\right)\right]\hat{\omega} \tag{2} $$

where the bias corrected body rotations are given by:

$$ \hat{\omega}^b_k = \tilde{\omega}^b_k - \hat{\mathbf{b}}_{g\,k}^{\;b} \tag{3} $$

Tracking of the platform alignment through zeroth order integration is considered sufficient for a first pass. First order integration with linear coning and sculling corrections could be a desirable later step, . Lastly, we renormalize the quaternion to unit norm when the condition $\mid 1 - \|{}^b_l\mathring{\mathbf{q}}_k\| \mid < 10^{-13}$ fails.

We proceed to translations. Body measured accelerations are resolved to the local navigation frame, and double integrated to velocities and positions. The midpoint rule is used, but may later be improved to trapezoidal rule, as used in [6]:

$$ \hat{\mathbf{a}}^b_k = \tilde{\mathbf{a}}^b_k - \hat{\mathbf{b}}_{a\,k}^{\;b} \tag{4} $$

$$ \begin{bmatrix} 0 \\ \hat{\mathbf{a}}^l_k \end{bmatrix} = {}^b_l\mathring{\mathbf{q}}^*_k \otimes \begin{bmatrix} 0 \\ \hat{\mathbf{a}}^b_k \end{bmatrix} \otimes {}^b_l\mathring{\mathbf{q}}_k \tag{5} $$

$$ \hat{\mathbf{f}}^l_k = \hat{\mathbf{a}}^l_k - \mathbf{g}^w \tag{6} $$

$$ \mathbf{P}^l_{k+1} = \mathbf{P}^l_k + \frac{\Delta t}{2}\left(\mathbf{V}^l_k + \mathbf{V}^l_{k-1}\right) \tag{7} $$

$$ \mathbf{V}^l_{k+1} = \mathbf{V}^l_k + \frac{\Delta t}{2}\left(\mathbf{f}^l_k + \mathbf{f}^l_{k-1}\right) \tag{8} $$

where we use the forward-left-up for all coordinate frames, making the world frame gravity vector $\mathbf{g}^w = \begin{bmatrix} 0 & 0 & +9.81 \end{bmatrix}^T$.

3.2 Inertial Navigation Error model – local frame rotations

Two of major formulations are present in literature. we repeat both here, as there are only subtle differences. We use a linearized state space model to propagate the error dynamics of the inertial navigation equations described above:

$$ \delta\dot{\mathbf{x}} = \mathbf{F}\delta\mathbf{x} + \mathbf{L}\eta \tag{9} $$

$$ \eta \sim \mathcal{N}(0, \mathbf{Q}) \tag{10} $$

where we introduce Gaussian process uncertainty η, and noise shaping matrix $\mathbf{L}$.

In this implementation, the error state of the INS is explicitly estimated. This in contrast with the implicit method used by [7], where the formulation is used to propagate the error state covariance to compute the Kalman gain.

The state estimate process described here stages this same process, so that the estimated error state is explicitly available. This is done so that error state updates to the INS can be throttled, rather than applying the Kalman update to the state directly. This gives the designer more control over the feedback operation of state estimate, and a much better handle on estimator performance when used in the feed forward debugging configuration.

The model proposed by Farrell, Titterton, Groves and Bekir [8, 9, 2, 10, 11, 12] is as follows: We use the error state vector

$$\delta\mathbf{x} = \begin{bmatrix}\delta\rho^l & \mathbf{b}_g^b & \delta\mathbf{V}^l & \delta\mathbf{P}^l & \mathbf{b}_a^b\end{bmatrix}^T \tag{11}$$

where the predicted local navigation frame is misaligned with the true local navigation frame by, roll-pitch-yaw, error angles $\delta\rho^l = \begin{bmatrix}\delta\phi & \delta\theta & \delta\psi\end{bmatrix}^T$, such that [8]:

$$\hat{\mathbf{R}}_l^b = \mathbf{R}_l^b\left(\mathbf{I} + \mathbf{P}\right) \tag{12}$$

$$\mathbf{P} = \lfloor \delta\rho \times \rfloor \tag{13}$$

$$\mathbf{P} = \begin{bmatrix} 0 & -\delta\psi & \delta\theta \\ \delta\psi & 0 & -\delta\phi \\ -\delta\theta & \delta\phi & 0 \end{bmatrix} \tag{14}$$

where $\lfloor \square \times \rfloor$ is the skew-symmetric matrix operator. We can also write the quaternion error rotation as:

$${}_l^b\mathring{\mathbf{q}} = {}_i^b\mathring{\mathbf{q}} \otimes {}_l^{\hat{l}}\delta\mathring{\mathbf{q}} \tag{15}$$

$${}_l^{\hat{l}}\delta\mathring{\mathbf{q}} = exmap\left(\delta\rho^l\right) \tag{16}$$

with *exmap* as the transform on the quaternion based rotational manifold.

$$\mathbf{F} = \begin{bmatrix} \boldsymbol{\Omega}_{iw}^w & -\hat{\mathbf{R}}\left({}_l^b\mathring{\mathbf{q}}^*\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tau_g & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\lfloor \hat{\mathbf{a}}^b \times \rfloor & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{R}}\left({}_l^b\mathring{\mathbf{q}}^*\right) \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tau_a \end{bmatrix} \tag{17}$$

The local frame predicted position error state is obtained with the linear measurement model:

$$\hat{\mathbf{y}} = \mathbf{H}\delta\mathbf{x} + \zeta \tag{18}$$

$$\zeta \sim \mathcal{N}\left(0, \mathbf{R}\right) \tag{19}$$

with $\mathbf{R}$ the usual Gaussian measurement covariance. We suspect a fairly large measurement covariance will be used, but the lower limit of this covariance will be the actual short term position estimate noise of from the kinematics based leg-odometry module.

The τ terms are heuristic terms which try predict that, on average, sensor bias evolution will be in the same direction. This assumption is included in the model, as a means to marginally reduce the required process noise on the bias states. These terms are generally only added once the rest of the model is operational and characterized.

The rotation of the world frame with respect to the inertial frame is given by

$$\boldsymbol{\Omega}_{iw}^w = \lfloor \omega_{iw}^w \times \rfloor \tag{20}$$

where the rotation of the earth at latitude, λ, is described as $\omega_{iw}^w = 15\ [^\circ/hr]\ \begin{bmatrix}\cos\left(\lambda\right) & 0 & \sin\left(\lambda\right)\end{bmatrix}^T$

Dominant process noise is introduced to only the sensor biases, as they are the main source of error in the system. Some small fraction of process noise is allocated to the other 9 states, given computational inaccuracies. We use the noise covariances, $\eta = \begin{bmatrix}\sigma_\rho^2 & \sigma_{b_g}^2 & \sigma_V^2 & \sigma_P^2 & \sigma_{b_a}^2\end{bmatrix}^T$, in shaping matrix:

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\hat{\mathbf{R}}\left({}^b_l\mathring{\mathbf{q}}^*\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{\mathbf{R}}\left({}^b_l\mathring{\mathbf{q}}^*\right) \end{bmatrix} \tag{21}$$

Another note is the importance of the initial covariance matrix. Just setting error state covariances along the diagonal of $\mathbf{P}_0$ to some large number is prone to failure. A good weighting between misalignment, sensor biases and translational velocity and position must be found, otherwise the EKF will diverge to unrecoverable linearized function errors.

3.3 Derivation of error models

3.3.1 Velocity error model – gravity feedback loop

We find the first order equation governing the propagation of the velocity error dynamics.

$$\dot{\hat{\mathbf{V}}}^l = {}^l_b\hat{\mathbf{R}}\,\hat{\mathbf{a}}^b - \hat{\mathbf{g}}^w \tag{22}$$

$$= (\mathbf{I} + \mathbf{P})\,{}^l_b\mathbf{R}\,\tilde{\mathbf{a}}^b - \hat{\mathbf{g}}^w \tag{23}$$

$$= {}^l_b\mathbf{R}\,\tilde{\mathbf{a}}^b + \mathbf{P}\,{}^l_b\mathbf{R}\,\tilde{\mathbf{a}}^b - \hat{\mathbf{g}}^w \tag{24}$$

where $\mathbf{P}$ represents the skew symmetric of the misalignment error angles. We introduce the accelerometer bias error as $\tilde{\mathbf{a}}^b = \mathbf{a}^b + \delta\mathbf{a}^b$.

$$\dot{\hat{\mathbf{V}}}^l = \dot{\mathbf{V}}^l + \delta\dot{\mathbf{V}}^l = {}^l_b\mathbf{R}\,\left(\mathbf{a}^b + \delta\mathbf{a}^b\right) + \mathbf{P}\,{}^l_b\mathbf{R}\,\left(\mathbf{a}^b + \delta\mathbf{a}^b\right) - \left(\mathbf{g}^w + \delta\mathbf{g}^w\right) \tag{25}$$

The error model therefore propagates as:

$$\delta\dot{\mathbf{V}}^l = {}^l_b\mathbf{R}\,\delta\mathbf{a}^b + \mathbf{P}\,{}^l_b\mathbf{R}\,\left(\mathbf{a}^b + \delta\mathbf{a}^b\right) - \delta\mathbf{g}^w \tag{26}$$

where the true state terms are taken as $\dot{\mathbf{V}}^l = {}^l_b\mathbf{R}\,\mathbf{a}^b - \mathbf{g}^w$. We drop the second order error term ${}^l_b\mathbf{R}\delta\mathbf{a}^b$,

$$\delta\dot{\mathbf{V}}^l \approx {}^l_b\mathbf{R}\,\delta\mathbf{a}^b + \mathbf{P}\,{}^l_b\mathbf{R}\,\mathbf{a}^b - \delta\mathbf{g}^w \tag{27}$$

$$\approx {}^l_b\mathbf{R}\,\delta\mathbf{a}^b + \mathbf{P}\,\mathbf{a}^l - \delta\mathbf{g}^w \tag{28}$$

$$\approx {}^l_b\mathbf{R}\,\delta\mathbf{a}^b - \lfloor\mathbf{a}^l\times\rfloor\,\delta\rho - \delta\mathbf{g}^w \tag{29}$$

We ignore the gravity error during the first pass implementation. Errors in our estimate of gravity will have to be absorbed by sensor bias estimates. Given that the robot will spend most of its time near level, we realize that gravity error will predominantly preside in the body vertical axis, and therefore exist as a z-axis accelerometer bias. Gravity error estimation can be further explored once a baseline performance system has been obtained.

3.3.2 Alternative formulation – body frame rotations

An alternative inertial error propagation model is presented by [7, 13]. Here misalignment error is estimated in the body, rather than local frame [14]:

$${}^b_l\mathbf{R} \approx \left(\mathbf{I}_3 - \lfloor{}^b\delta\rho\times\rfloor\right)\,{}^b_l\hat{\mathbf{R}} \tag{30}$$

which is simply the inverse of the local frame error angles proposed in eq. (12):

$$
\begin{aligned}
{}^{l}_{b}\mathbf{R} &\approx {}^{l}_{b}\hat{\mathbf{R}} \left(\mathbf{I}_3 - \lfloor {}^{b}\delta\rho \times \rfloor\right)^T && (31)\\
&\approx {}^{l}_{b}\hat{\mathbf{R}} \left(\mathbf{I}_3 + \lfloor {}^{b}\delta\rho \times \rfloor\right) && (32)
\end{aligned}
$$

this shows the difference between the two models – such that ${}^{b}\delta\rho = -{}^{l}\delta\rho$. It is important to stress the wander-azimuth angle must be considered when local and world frame mechanizations are used in the same system. If we try to estimate the wander-azimuth alignment between the world and local frames, we need to consider how we define the misalignment angles.

$$
\mathbf{F} = \begin{bmatrix}
\boldsymbol{\Omega}^{w}_{iw} & -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
-\hat{\mathbf{R}}\left({}^{b}_{l}\mathring{\mathbf{q}}^{*}\right) \lfloor \hat{\mathbf{a}}^{b} \times \rfloor & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\hat{\mathbf{R}}\left({}^{b}_{l}\mathring{\mathbf{q}}^{*}\right) \\
\mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix} \tag{33}
$$

And associated noise shaping matrix:

$$
\mathbf{L} = \begin{bmatrix}
\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & -\hat{\mathbf{R}}\left({}^{b}_{l}\mathring{\mathbf{q}}^{*}\right) & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3
\end{bmatrix} \tag{34}
$$

3.4 Measurement and Aiding model

We envision either position or velocity aiding to the inertial solution. Both leg-kinematics and visual odometry can be used as position or velocity aides. Velocity aiding would decouple DC errors of aiding source, but may result in slow drift during static operation. During static operation – that is where new footsteps are not made – positional aiding can be used. This can be achieved with a switching architecture.

4 Leg Odometry from Kinematics

4.1 Leg Kinematics

We describe the leg kinematics using 4×4 transform matrix, consisting of:

$$
\mathbf{T} = \begin{bmatrix} {}^{b}_{a}\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{35}
$$

From experience in the VRC, and practical observation, we choose to ignore the pitch and roll joint encoders for state-estimation – at the cost of requiring an separate orientation estimate. During the VRC we found that small errors in the ankle joints, would result in large positional variations in pelvis position. We would like to use the orientation estimate form inertial measurements. This modifies the leg-kinematic to one where the robot is assumed to be walking on stubs, rather than surface following feet.

Pelvis position in the local frame, l, can be obtained by accumulating two rigid transforms. With b denoting body reference frame. The first is from the origin to the standing foot, and then foot to

pelvis through kinematics. Ignoring the rotations of the foot, f:

$$\mathbf{T}_b^l = \mathbf{T}_f^l \otimes \mathbf{T}_b^f \tag{36}$$

$$= \begin{bmatrix} {}_b^l\mathbf{R} & \mathbf{t}_{l\to f}^l \\ \mathbf{0} & 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{I} & \mathbf{t}_{f\to b}^b \\ \mathbf{0} & 1 \end{bmatrix} \tag{37}$$

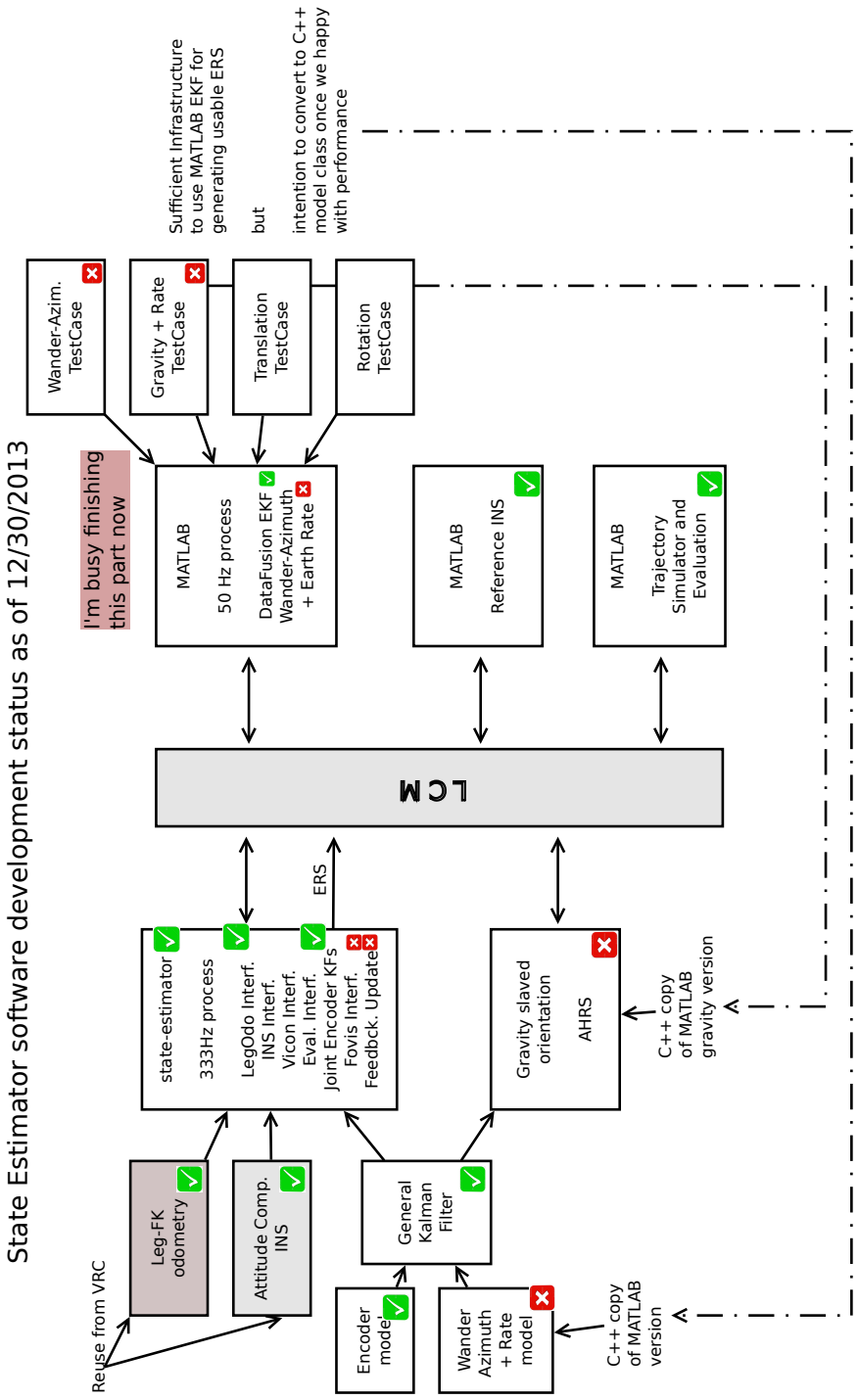
$$= \begin{bmatrix} {}_b^l\mathbf{R} & \left({}_b^l\mathbf{R}\,\mathbf{t}_{f\to b}^b + \mathbf{t}_{l\to f}^l\right) \\ \mathbf{0} & 1 \end{bmatrix} \tag{38}$$

$$= \begin{bmatrix} {}_b^l\mathbf{R} & \left(\mathbf{t}_{f\to b}^l + \mathbf{t}_{l\to f}^l\right) \\ \mathbf{0} & 1 \end{bmatrix} \tag{39}$$

$$= \begin{bmatrix} {}_b^l\mathbf{R} & \mathbf{t}_{l\to b}^l \\ \mathbf{0} & 1 \end{bmatrix} \tag{40}$$

This is what is currently implemented in the `TwoLegOdometry` class, following the VRC.

5 Software Status



5.1 Foot transition classifier and special cases

Differential force, Schmidtt trigger and time delay.

5.2 Handling failure cases

- Application to catering for leg-kinematic failure cases. Such as falling over.
- How we shape measurement noise matrix, based on how much force is seen on the feet.
- When hanging from the rope, we use large variance static position update.
- When feet are not in good contact, such as falling, we just ignore leg-odometry measurements. Wait for motion to stop moving and reset the leg kinematics with the INS based solution. We can employ large variance zero velocity updates, until robot starts moving again.

References

[1] X. Kong, *Inertial navigation system algorithms for low cost IMU*. PhD thesis, Department of Mechanical and Mechatronic Engineering, Graduate School of Engineering, University of Sydney, 2000.

[2] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, GNSS Technology and Application Series, 2008.

[3] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *Journal of guidance, control, and dynamics*, vol. 21, no. 1, pp. 19–28, 1998.

[4] K. R. Britting, *Inertial Navigation Systems Analysis*. 1971.

[5] A. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, vol. 174. AIAA, 1997.

[6] P. Zarchan and H. Musoff, *Fundamentals of Kalman filtering: a practical approach*, vol. 208. AIAA, 2005.

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3565–3572, IEEE, 2007.

[8] J. Farrell, *Aided navigation: GPS with high rate sensors*. McGraw-Hill New York, 2008.

[9] D. Titterton and J. Weston, *Strapdown inertial navigation technology*, vol. 17. IET, 2004.

[10] E. Bekir, *Introduction to modern navigation systems*. World Scientific, 2007.

[11] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*. Wiley. com, 2011.

[12] J. Farrell and M. Barth, *The global positioning system and inertial navigation*, vol. 61. McGraw-Hill New York, 1999.

[13] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. L. Speyer, "Observability of error states in gps/ins integration," *Vehicular Technology, IEEE Transactions on*, vol. 54, no. 2, pp. 731–743, 2005.

[14] M. Li and A. I. Mourikis, "Improving the accuracy of ekf-based visual-inertial odometry," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 828–835, IEEE, 2012.