Hybrid Unscented Kalman Filter: Application to the Simplest Walker ⋆

James Bittler, Pranav A. Bhounsule

Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, 842 W Taylor St, Chicago, IL 60607, USA. (e-mail: jbittler@uic.edu; pranav@uic.edu)

.

Abstract: State estimation of hybrid dynamic systems, such as legged robots, is challenging because of the presence of non-smooth dynamics. This paper applies the Unscented Kalman Filter (UKF) state estimator and two novel hybrid extensions (HUKF) to a hybrid system, the simplest walking model. These estimators are identical far from the switching boundary, which partitions dynamic domains, but apply different time update algorithms at the switching boundary. (1) UKF permits sigma points to propagate through the system's hybrid dynamics. (2) HUKF-SPG (Sigma Point Generation) generates new sigma points when the weighted mean of the initial sigma points is on the switching boundary. (3) HUKF-SPT (Sigma Point Transformation) transforms the sigma points forward and backward in time through the system's hybrid dynamics only when the weighted mean of the initial sigma points is on the switching boundary. Results here shows that HUKF-SPG and HUKF-SPT have a lower absolute error but modestly more computations compared to UKF. A caveat of HUKF-SPT is it can only apply to conservative systems. HUKF-SPG is more general and could be applied to any hybrid system.

Keywords: Estimation and filtering, Hybrid and switched systems modeling, Mobile robots, Sensing, Legged Systems.

1. INTRODUCTION

Recent advances in hardware and control techniques are driving a fresh wave of robotic applications based on the ability of legged systems to traverse complex terrain (e.g., Raibert (2010)). However, state estimation for legged systems has received less attention than system control. The hybrid nature of legged systems involving switching boundaries (e.g., foot striking the ground) and changing dynamics (e.g., bipedal robot transition from single stance to double stance) makes state estimation challenging. Improvement to state estimation will enable higher fidelity control, driving increasingly more sophisticated applications in the future.

State observers can provide an estimate of the full state based on measurements of some states. These usually work well when sensor noise is small. Grizzle et al. (2007) have shown if an exponentially stable feedback controller exists, and observer error converges to zero sufficiently rapidly, then a periodic system can be exponentially stabilized by the controller based on the observer. When the system is nonlinear and the noise is substantial, one can use an extended Kalman filter (EKF) for estimation. In the EKF, the linearization of the system dynamics and measurement model is used for state estimation. We may use the EKF with a kinematic model (Bloesch et al. (2013)) or a dynamics model (Kong et al. (2021); Hartley et al. (2018)) for full pose estimation. There are two issues with the EKF. The first issue is that the switching boundary causes nonlinearity in the dynamics that an EKF cannot handle. This has been addressed by using the salted Kalman filter (Kong et al. (2021)). Here, the saltation matrix preserves the second moments across the switching boundary. The second issue is that the error dynamics of the Kalman filter are state dependent which causes convergence issues. This has been addressed by developing an invariant extended Kalman filter (Hartley et al. (2018)). The saltation matrix can be combined with the invariant extended Kalman filter to achieve a resolution of both issues (Gao et al. (2021)).

The unscented Kalman filter (UKF) is another method for state estimation of a nonlinear system (Julier et al. (2000)). Unlike EKF which is accurate to the first-order linearization of the system, UKF carefully choses sampling points, known as sigma points (SPs), to estimate the mean and covariance which are accurate to third order in a Taylor series expansion (Wan and Van Der Merwe (2000)). This method is referred to as the Unscented Transformation (UT) (Julier and Uhlmann (2004)). In this paper, our primary contribution is the modification of the UT within the UKF for state estimation of a hybrid system – the Hybrid Unscented Kalman Filter (HUKF). The HUKF is identical to the UKF unless the system reaches the switching boundary. At the switching boundary, we investigate three methods, one standard and two novel, to update the filter. We present test results on the benchmark model of walking (Bhounsule (2014)).

⋆ The work was supported by NSF grant 2128568.

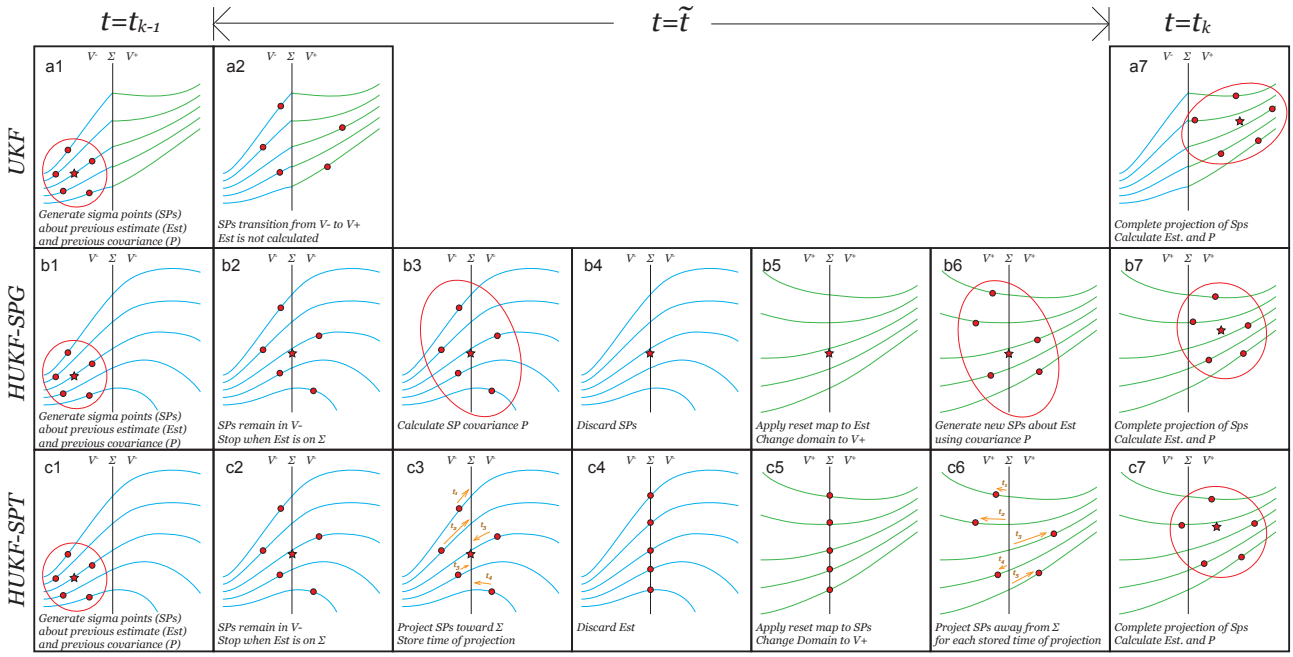

Fig. 1. Illustration of the Unscented Kalman Filter time update steps; (a) standard UKF, (b) HUKF-SPG, (c) HUKF-SPT. These differ in how the switching boundary (black vertical line, Σ) is traversed from the initial domain (blue field, V^-) to the final domain (green field, V^+). (a) permits the sigma points (red dots, SPs) to propagate trough V^-, across Σ, and into V^+, whereby a mean (red star, $\hat{\boldsymbol{x}}$) and covariance (red ellipse, $\hat{\mathbf{P}}$) are calculated. (b) propagates SPs through V^- until $\hat{\boldsymbol{x}}$ is on Σ, where $\hat{\mathbf{P}}$ is calculated. The reset map is applied to the $\hat{\boldsymbol{x}}$, new SPs generated, and new SPs propagated through V^+. (c) propagates SPs through V^- until $\hat{\boldsymbol{x}}$ is on Σ, where each SP is propagated to Σ in V^-, reset map is applied, each SP is propagated away from Σ in V^+, and transformed SPs complete propagation in V^+.

2. UNSCENTED KALMAN FILTER ALGORITHMS FOR HYBRID SYSTEMS

2.1 System Dynamics

The complete set of equations are summarized below:

$$t = t_0: \quad \boldsymbol{x}(t_0) = \boldsymbol{x_0} \tag{1}$$

$$t_0 \le t < \tilde{t}: \quad \dot{\boldsymbol{x}} = \boldsymbol{f}^-(\boldsymbol{x}), \quad S(\boldsymbol{x}(t)) < 0, \quad \boldsymbol{x} \in V^- \tag{2}$$

$$t = \tilde{t}: \quad S(\boldsymbol{x}(\tilde{t})) = 0, \quad \boldsymbol{x} \in \Sigma \tag{3}$$

$$t = \tilde{t}: \quad \boldsymbol{x}^+(\tilde{t}) = \mathcal{R}(\boldsymbol{x}(\tilde{t})) \tag{4}$$

$$t > \tilde{t}: \quad \dot{\boldsymbol{x}} = \boldsymbol{f}^+(\boldsymbol{x}), \quad S(\boldsymbol{x}(t)) > 0, \quad \boldsymbol{x} \in V^+, \tag{5}$$

Furthermore, the system is discretized; subscript k indicates a discrete timestep t_k. The system dynamics given by Eqs. 2 and 5 are written as, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{f}^{\pm}(\boldsymbol{x}_k, \boldsymbol{w}_k)dt$, where $\boldsymbol{w}_k$ is discrete process noise that is held constant between time steps. Process noise is assumed to be normally distributed zero mean, $E[\boldsymbol{w}_k] = 0$, with variance $E[\boldsymbol{w}_k \boldsymbol{w}_k^{\mathrm{T}}] = \boldsymbol{Q}$.

2.2 System Measurement

Discrete measurement equation is written as $\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{v}_k)$, where $\boldsymbol{v}_k$ is the measurement noise, and is zero mean, $E[\boldsymbol{w}_k] = E[\boldsymbol{v}_k] = 0$, with variance $E[\boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}] = \boldsymbol{R}$. System and process noise are independent, $E[\boldsymbol{w}_k \boldsymbol{v}_k^{\mathrm{T}}] = \mathbf{0}$. It is assumed that there is no noise on the switching boundary and the reset condition, w.l.o.g.

2.3 Unscented Kalman Filter Applied To Hybrid Systems

The Unscented Kalman Filter was applied to ball bouncing on a wall by Julier and Uhlmann (2004) (pp. 415). We adapt the same algorithm, but use the notation in Wan and Van Der Merwe (2000) to present the pseudo code in Table 1. The UKF has five primary steps: initialization and generation of SPs (lines 1-4), propagation of the SPs through the dynamics (line 5) referred to as the time update, measurement of SPs (lines 6-8), calculation of Kalman gain (lines 9-12), and use of the measurements to update the state and covariance (lines 13-14).

Time updates which meet or traverse the switching boundary require special management. The pseudocode for the time update is shown in Table 2 and is illustrated in Fig 1 a1-a7. As shown in a2, when propagating from t_{k-1} to t_k, the i^{th} SP crosses the switching boundary at some time $\tilde{t}_i$, where $\tilde{t}_{k-1} < \tilde{t}_i \le \tilde{t}_k$. A root finding algorithm can be used to determine the value of $\tilde{t}$.

2.4 Hybrid Unscented Kalman Filter with Sigma Point Generation (HUKF-SPG)

The Hybrid Unscented Kalman Filter with Sigma Point Generation (HUKF-SPG) follows the same general approach as UKF (Table 1) with exception of the time update near the switching boundary.

The pseudocode for the time update is shown in Table 3 and illustrated in Fig 1, b1-b7. As shown in b2, individual

SPs are not subject to the system's hybrid dynamics, but projected forward in $\boldsymbol{f}^-(\cdot)$ to some time $\tilde{t}$ such that the weighted mean of the SPs at that time $\tilde{\mathbf{x}}^-$ (line 4) lies exactly on the switching boundary, i.e. $S(\tilde{\mathbf{x}}^-) = 0$. This mean is represented graphically as a red star. As with the UKF, a root finding algorithm can be used to determine the value of $\tilde{t}$.

With mean on the switching boundary, as illustrated in b3, the covariance $\tilde{\mathbf{P}}_{xx}^-$ is calculated (line 5). Thereafter, the SPs are discarded as shown in b4. The reset map is then applied to the mean $\tilde{\mathbf{x}}^-$, resulting in updated mean $\tilde{\mathbf{x}}^+$ (line 9). This is illustrated in b5 with identity reset map for sake of clarity but w.l.o.g. The covariance then is used to generate new SPs (line 10), as shown in b6. Finally, as illustrated in b7, these new SPs are projected in $\boldsymbol{f}^+(\cdot)$ from $\tilde{t}$ to t_k, at which time the weighted mean $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_{xx}$ are calculated.

This filter is named HUKF with Sigma Point Generation because, as shown in b6, the mean transformed by the reset map, $\tilde{\mathbf{x}}^+$, is used in conjunction with the covariance, $\tilde{\mathbf{P}}_{xx}^-$, to generate new SPs.

2.5 Hybrid Unscented Kalman Filter with Sigma Point Transformation (HUKF-SPT)

The Hybrid Unscented Kalman Filter with Sigma Point Transformation (HUKF-SPT) follows the same general approach as UKF (Table 1) except for the time update near the switching boundary.

The pseudocode for the time update is shown in Table 4 and explained pictorially in Fig 1, c1 to c7. As illustrated in c2, individual SPs are not subject to the system's hybrid dynamics, but projected forward in $\boldsymbol{f}^-(\cdot)$ to some time $\tilde{t}$ such that the weighted mean of the SPs at that time $\tilde{\mathbf{x}}^-$ (line 4) lies exactly on the switching boundary, i.e. $S(\tilde{\mathbf{x}}^-) = 0$. This mean is represented graphically as a red star. As with the UKF, a root finding algorithm can be used to determine the value of $\tilde{t}$.

With mean on the switching boundary, as illustrated in c3, each SP, $\tilde{\mathcal{X}}_i^-$, is individually projected in $\boldsymbol{f}^-(\cdot)$ from $\tilde{t}$ to some time $\bar{t}_i$ (line 11) such that its resultant value, $\bar{\mathcal{X}}_i^-$, lies exactly on the switching boundary, i.e. $S(\bar{\mathcal{X}}_i^-) = 0$. Note that for SPs which are beyond the switching boundary at $\tilde{t}$, $\bar{t}_i < \tilde{t}$, i.e. those SPs must be projected backward in time.

With all SPs exactly on the switching boundary, the mean is discarded and the reset map is applied to each SP $\bar{\mathcal{X}}_i^-$ (line 13), resulting in updated SPs, $\bar{\mathcal{X}}_i^+$. This is illustrated in c4 and c5. These SPs are then individually projected in $\boldsymbol{f}^+(\cdot)$ from $\bar{t}_i$ to time $\tilde{t}$ (line 15). This again requires projection backward in time as shown in c6. Finally, as illustrated in c7, the transformed SPs are projected in $\boldsymbol{f}^+(\cdot)$ from $\tilde{t}$ to t_k, at which time the weighted mean $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_{xx}$ are calculated.

This filter is named HUKF with Sigma Point Transformation because, as shown in c3 and c6, the SPs are transformed by the system's hybrid dynamics.

Table 1 Unscented Kalman Filter

in: pre. estimate $\hat{\hat{\mathbf{x}}}_{k-1}$; cur. measurement $\mathbf{y}_k$; pre. cov. $\hat{\hat{\mathbf{P}}}_{k-1}$; process noise cov. $\mathbf{Q}$; measurement noise cov. $\mathbf{R}$; parameters α, β, κ

out: cur. estimate $\hat{\hat{\mathbf{x}}}_k$; cur. cov. $\hat{\hat{\mathbf{P}}}_k$

% Generate SPs (see line 15)

1: $\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^w, \mathcal{X}_{k-1}^v = \mathbf{genSPs}(\hat{\hat{\mathbf{x}}}_{k-1}, \hat{\hat{\mathbf{P}}}_{k-1}, \mathbf{Q}, \mathbf{R}, \alpha, \kappa)$

% Generate $2L+1$ weights

2: $\mathbf{W} = \lambda/(2(L+\lambda)) * \mathbf{ones}(1, 2L)$

3: $\mathbf{W}^m = [\lambda/(L+\lambda), \mathbf{W}]$ ▷ *for mean*

4: $\mathbf{W}^c = [\lambda/(L+\lambda) + (1-\alpha^2+\beta), \mathbf{W}]$ ▷ *for covariance*

% Time Update differs as indicated below
% UKF is in Table 2
% HUKF-SPG is in Table 3
% HUKF-SPT is in Table 4

5: $\hat{\mathbf{x}}_k, \hat{\mathcal{X}}_k, \hat{\mathbf{P}}_{xx} = \mathbf{time\_update}(\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^w, \mathbf{W}^m, \mathbf{W}^c)$

% Propagate SPs through measurements

6: **for** i=1:(2L+1) **do**

7: $\quad \hat{\mathcal{Y}}_{k,i} = \boldsymbol{h}(\hat{\mathcal{X}}_{k,i}, \mathcal{X}_{k-1,i}^v)$

8: **end for**

% Calculate the mean and covariance

9: $\hat{\mathbf{y}}_k = \sum_{i=1}^{2L+1} W_i^m \hat{\mathcal{Y}}_{k,i}$

10: $\hat{\mathbf{P}}_{yy} = \sum_{i=1}^{2L+1} W_i^c [\hat{\mathcal{Y}}_{k,i} - \hat{\mathbf{y}}_k][\hat{\mathcal{Y}}_{k,i} - \hat{\mathbf{y}}_k]^T$

% Calculate cross-covariance and Kalman gain

11: $\hat{\mathbf{P}}_{xy} = \sum_{i=1}^{2L+1} W_i^c [\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}_k][\hat{\mathcal{Y}}_{k,i} - \hat{\mathbf{y}}_k]^T$

12: $\mathcal{K} = \hat{\mathbf{P}}_{xy} \hat{\mathbf{P}}_{yy}^{-1}$

% Measurement update

13: $\hat{\hat{\mathbf{x}}}_k = \hat{\mathbf{x}}_k + \mathcal{K}(\hat{\mathbf{y}}_k - \mathbf{y}_k)$

14: $\hat{\hat{\mathbf{P}}}_k = \hat{\mathbf{P}}_{xx} - \mathcal{K} \hat{\mathbf{P}}_{yy} \mathcal{K}^T$

15: **function** GENSPS($\hat{\hat{\mathbf{x}}}_{k-1}, \hat{\hat{\mathbf{P}}}_{k-1}, \mathbf{Q}, \mathbf{R}, \alpha, \kappa$)

% Initialize the augmented state and covariance

16: $\quad \mathbf{x}_{k-1}^a = [\hat{\hat{\mathbf{x}}}_{k-1}^T, \mathbf{0}_w^T, \mathbf{0}_v^T]^T$ ▷ $\mathbf{0}_w \in \mathbb{R}^{n\times 1}$, $\mathbf{0}_v \in \mathbb{R}^{m\times 1}$

17: $\quad \mathbf{P}^a = \mathbf{diag}(\hat{\hat{\mathbf{P}}}_{k-1}, \mathbf{Q}, \mathbf{R})$

% Generate $2L+1$ SPs

18: $\quad \lambda = \alpha^2(L+\kappa)$

19: $\quad \mathbf{A} = \sqrt{(L+\lambda)}\sqrt{\mathbf{P}_{k-1}^a}$ ▷ $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^T$

20: $\quad \mathcal{X}_{k-1}^a = [\mathbf{x}_{k-1}^a,\ \mathbf{x}_{k-1}^a + \mathbf{A},\ \mathbf{x}_{k-1}^a - \mathbf{A}]$

21: $\quad \mathcal{X}_{k-1}^x = \mathcal{X}_{k-1}^a(1{:}n,\ :)$

22: $\quad \mathcal{X}_{k-1}^w = \mathcal{X}_{k-1}^a(n+1{:}2n,\ :)$

23: $\quad \mathcal{X}_{k-1}^v = \mathcal{X}_{k-1}^a(2n+1{:}L,\ :)$

$\quad$ **return** $\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^w, \mathcal{X}_{k-1}^v$

24: **end function**

3. RESULTS

3.1 Hybrid system: The simplest walker

In this study, state estimation of a walking model known as the simplest walker (Garcia et al. (1998)) is investigated. The simplest walker, shown in Fig. 2, is able to walk down a shallow ramp without any control when launched with the correct initial conditions.

The model has a point mass M at the hip, point feet of mass m, leg length is ℓ, gravity is g, and ramp slope

Table 2 Time Update: UKF

in: SPs of pre. state $\mathcal{X}^x_{k-1}$; SPs of pre. process noise $\mathcal{X}^w_{k-1}$; weight for mean $\mathbf{W}^m$; weight for cov. $\mathbf{W}^c$
out: cur. priori est. $\hat{\mathbf{x}}_k$; cur. SPs $\hat{\mathcal{X}}_k$; cur. priori cov. $\hat{\mathbf{P}}_{xx}$

1: **for** i=1:(2L+1) **do**
% Project SPs from t_{k-1} to $\tilde{t}$
*% such that ($\tilde{t} = t_k$ **and** $S(\tilde{\mathcal{X}}_i^-) \leq 0$) **or***
*% ($\tilde{t} < t_k$ **and** $S(\tilde{\mathcal{X}}_i^-) = 0$)*
2: $\tilde{\mathcal{X}}_i^- = \mathcal{X}^x_{k-1,i} + \int_{t_{k-1}}^{\tilde{t}} \boldsymbol{f}^-(\mathcal{X}^x_{k-1,i}, \mathcal{X}^w_{k-1,i})dt$
% Manage domain change
3: **if** ($\tilde{t} = t_k$ **and** $S(\tilde{\mathcal{X}}_i^-) < 0$) **then**
4: $\hat{\mathcal{X}}_{k,i} = \tilde{\mathcal{X}}_i^-$
5: **else**
% Apply reset map
6: $\tilde{\mathcal{X}}_i^+ = \mathcal{R}(\tilde{\mathcal{X}}_i^-)$
7: **if** $\tilde{t} = t_k$ **then** $\triangleright\ \tilde{t} = t_k \Leftrightarrow S(\tilde{\mathcal{X}}_i^-) = 0$
8: $\hat{\mathcal{X}}_{k,i} = \tilde{\mathcal{X}}_i^+$
9: **else**
% Complete SP projection
10: $\hat{\mathcal{X}}_{k,i} = \tilde{\mathcal{X}}_i^+ + \int_{\tilde{t}}^{t_k} \boldsymbol{f}^+(\tilde{\mathcal{X}}_i^+, \mathcal{X}^w_{k-1,i})dt$
11: **end if**
12: **end if**
13: **end for**
% Calculate mean and covariance
14: $\hat{\mathbf{x}}_k = \sum_{i=1}^{2L+1} \mathbf{W}_i^m \hat{\mathcal{X}}_{k,i}$
15: $\hat{\mathbf{P}}_{xx} = \sum_{i=1}^{2L+1} \mathbf{W}_i^c [\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}_k][\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}_k]^T$

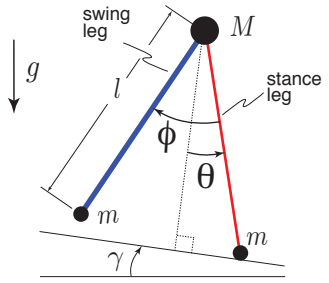

Fig. 2. The simplest walking model analyzed by Garcia et al. (1998) was used to test the filter algorithms.

is γ. Garcia et al. (1998) did two simplifications: non-dimensionalization of time with $\sqrt{g/\ell}$, and analysis of the limiting case, $m/M \to 0$. The net effect of these two simplifications is that the equations of motion have only one free parameter, slope γ. The system state is $\boldsymbol{x} = [\theta, \dot{\theta}, \phi, \dot{\phi}]^T$. The switching condition is $S(\boldsymbol{x}) = 2\theta - \phi$. The reset map is $\mathcal{R}(\boldsymbol{x}) = [\theta,\ \{-1 + \cos(2\theta)\}\cos(2\theta)\dot{\theta},\ \phi,\ -\cos(2\theta)\dot{\theta}]^T$. The initial dynamics are $\boldsymbol{f}^-(\boldsymbol{x}) = [\dot{\theta},\ \sin(\theta - \gamma),\ \dot{\phi},\ \sin(\theta - \gamma) + \{\dot{\theta}^2 - \cos(\theta - \gamma)\}\sin(\phi)]^T$. The final dynamics are $\boldsymbol{f}^+(\boldsymbol{x}) = [\dot{\theta} - \dot{\phi},\ \sin(\phi - \theta + \gamma) + (\dot{\phi}^2 - \cos(\phi - \theta + \gamma))\sin(\phi)\dot{\theta},\ \sin(\phi - \theta + \gamma)]^T$

The simplest walker is sensitive to the initial conditions; random initial conditions are not able to generate feasible walking solutions. We use fixed point analysis to generate feasible initial conditions (see Strogatz (2018)). This is based on the Poincaré section, an instant in the motion

Table 3 Time Update: HUKF-SPG

in: SPs of pre. state $\mathcal{X}^x_{k-1}$; SPs of pre. process noise $\mathcal{X}^w_{k-1}$; weight for mean $\mathbf{W}^m$; weight for cov. $\mathbf{W}^c$; process noise cov. $\mathbf{Q}$; measurement noise cov. $\mathbf{R}$; params. α, β, κ
out: cur. priori est. $\hat{\mathbf{x}}_k$; cur. SPs $\hat{\mathcal{X}}_k$; cur. priori cov. $\hat{\mathbf{P}}_{xx}$

% Project SPs from t_{k-1} to $\tilde{t}$, calculate mean $\tilde{\mathbf{x}}^-$
*% such that ($\tilde{t} = t_k$ **and** $S(\tilde{\mathbf{x}}^-) \leq 0$) **or***
*% ($\tilde{t} < t_k$ **and** $S(\tilde{\mathbf{x}}^-) = 0$)*
1: **for** i=1:(2L+1) **do**
2: $\tilde{\mathcal{X}}_i^- = \mathcal{X}^x_{k-1,i} + \int_{t_{k-1}}^{\tilde{t}} \boldsymbol{f}^-(\mathcal{X}^x_{k-1,i}, \mathcal{X}^w_{k-1,i})dt$
3: **end for**
4: $\tilde{\mathbf{x}}^- = \sum_{i=1}^{2L+1} \mathbf{W}_i^m \tilde{\mathcal{X}}_i^-$
5: $\tilde{\mathbf{P}}_{xx}^- = \sum_{i=1}^{2L+1} \mathbf{W}_i^c [\tilde{\mathcal{X}}_i^- - \tilde{\mathbf{x}}^-][\tilde{\mathcal{X}}_i^- - \tilde{\mathbf{x}}^-]^T$
% Manage domain change
6: **if** ($\tilde{t} = t_k$ **and** $S(\tilde{\mathbf{x}}^-) < 0$) **then**
7: $\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}^-, \quad \hat{\mathcal{X}}_k = \tilde{\mathcal{X}}^-, \quad \hat{\mathbf{P}}_{xx} = \tilde{\mathbf{P}}_{xx}^-$
8: **else**
% Apply reset map
9: $\tilde{\mathbf{x}}^+ = \mathcal{R}(\tilde{\mathbf{x}}^-)$
% Generate new SPs (see table 2, line 15)
10: $\tilde{\mathcal{X}}^+, \_, \_ = \mathbf{genSPs}(\tilde{\mathbf{x}}^+, \tilde{\mathbf{P}}_{xx}^-, \mathbf{Q}, \mathbf{R}, \alpha, \kappa)$
11: **if** $\tilde{t} = t_k$ **then** $\triangleright\ \tilde{t} = t_k \Leftrightarrow S(\tilde{\mathbf{x}}^-) = 0$
12: $\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}^+, \quad \hat{\mathcal{X}}_k = \tilde{\mathcal{X}}^+, \quad \hat{\mathbf{P}}_{xx} = \tilde{\mathbf{P}}_{xx}^-$
13: **else**
% Complete SP projection
14: **for** i=1:(2L+1) **do**
15: $\hat{\mathcal{X}}_{k,i} = \tilde{\mathcal{X}}_i^+ + \int_{\tilde{t}}^{t_k} \boldsymbol{f}^+(\tilde{\mathcal{X}}_i^+, \mathcal{X}^w_{k-1,i})dt$
16: **end for**
% Calculate mean and covariance
17: $\hat{\mathbf{x}}_k = \sum_{i=1}^{2L+1} \mathbf{W}_i^m \hat{\mathcal{X}}_{k,i}$
18: $\hat{\mathbf{P}}_{xx} = \sum_{i=1}^{2L+1} \mathbf{W}_i^c [\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}][\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}]^T$
19: **end if**
20: **end if**

cycle and the Poincaré map, a function that maps an initial condition from one Poincaré section to the next. We choose the Poincaré section to be at the instant after the front leg is in contact with the ground and the rear leg is about to move, identified as the instant after foot-strike. We assume an initial condition $\boldsymbol{x}_0$ at this instant. Then we integrate the equations of motion $\dot{\boldsymbol{x}} = \boldsymbol{f}^-(\boldsymbol{x}, \boldsymbol{0})$ until the switching boundary is reached, i.e. $S(\boldsymbol{x}) = 0$. Then we apply the reset map $\mathcal{R}(\boldsymbol{x})$. These steps provide the Poincaré map, $\boldsymbol{M}(\boldsymbol{x})$, such that $\boldsymbol{x}_1 = \boldsymbol{M}(\boldsymbol{x}_0)$. Using a root finder, we search for $\boldsymbol{x}_1 = \boldsymbol{x}_0 = \boldsymbol{x}^\star$ such that $\boldsymbol{x}^\star - \boldsymbol{M}(\boldsymbol{x}^\star) = 0$. For the free parameter, $\gamma = 0.009$, this gives the following fixed point $\boldsymbol{x}^\star = [0.2003\ -0.1998\ 0.4006\ -0.0158]$ corresponding to a step time $t_{\text{step}} = 3.8824$ (see Bhounsule (2014) for more details and code).

3.2 Simulation and filter parameters

Only behavior of the UKF and HUKF variants about the switching boundary is of interest in this study. For this reason, and to reduce computational load, the system was first initialized at the deterministic fixed point $\boldsymbol{f}^-(\boldsymbol{x}^\star, \boldsymbol{0})$, and integrated forward for 3 sec. yielding the

Table 4 Time Update: HUKF-SPT

in: SPs of pre. state $\mathcal{X}^x_{k-1}$; SPs of pre. process noise $\mathcal{X}^w_{k-1}$; weight for mean $\mathbf{W}^m$; weight for cov. $\mathbf{W}^c$;
out: cur. priori est. $\hat{\mathbf{x}}_k$; cur. SPs $\hat{\mathcal{X}}_k$; cur. priori cov. $\hat{\mathbf{P}}_{xx}$

% Project SPs from t_{k-1} to $\tilde{t}$, calculate mean $\tilde{\mathbf{x}}^-$
*% such that ($\tilde{t} = t_k$ **and** $S(\tilde{\mathbf{x}}^-) \leq 0$) **or***
*% ($\tilde{t} < t_k$ **and** $S(\tilde{\mathbf{x}}^-) = 0$)*

for i=1:(2L+1) **do**
  $\tilde{\mathcal{X}}_i^- = \mathcal{X}^x_{k-1,i} + \int_{t_{k-1}}^{\tilde{t}} \boldsymbol{f}^-(\mathcal{X}^x_{k-1,i}, \mathcal{X}^w_{k-1,i})dt$
end for
$\tilde{\mathbf{x}}^- = \sum_{i=1}^{2L+1} \mathbf{W}_i^m \tilde{\mathcal{X}}_i^-$
$\tilde{\mathbf{P}}^-_{xx} = \sum_{i=1}^{2L+1} \mathbf{W}_i^c [\tilde{\mathcal{X}}_i^- - \tilde{\mathbf{x}}^-][\tilde{\mathcal{X}}_i^- - \tilde{\mathbf{x}}^-]^T$

% Manage domain change
if ($\tilde{t} = t_k$ **and** $S(\tilde{\mathbf{x}}^-) < 0$) **then**
  $\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}^-, \quad \hat{\mathcal{X}}_k = \tilde{\mathcal{X}}^-, \quad \hat{\mathbf{P}}_{xx} = \tilde{\mathbf{P}}^-_{xx}$
else
  % Project SPs from $\tilde{t}$ to $\bar{t}_i$, such that $S(\bar{\mathcal{X}}_i^-) = 0$,
  % where $\bar{t}_i < \tilde{t}$ if $S(\bar{\mathcal{X}}_i^-) > 0$
  for i=1:(2L+1) **do**
    % Project SP toward switching boundary
    $\bar{\mathcal{X}}_i^- = \tilde{\mathcal{X}}_i^- + \int_{\tilde{t}}^{\bar{t}_i} \boldsymbol{f}^-(\tilde{\mathcal{X}}_i^-, \mathcal{X}^w_{k-1,i})\, dt$
    % Apply reset map
    $\bar{\mathcal{X}}_i^+ = \mathcal{R}(\bar{\mathcal{X}}_i^-)$
    % Project SP away from switching boundary
    $\tilde{\mathcal{X}}_i^+ = \bar{\mathcal{X}}_i^+ + \int_{\bar{t}_i}^{\tilde{t}} \boldsymbol{f}^-(\bar{\mathcal{X}}_i^+, \mathcal{X}^w_{k-1,i})\, dt$
  end for

  if $\tilde{t} < t_k$ **then**
    % Complete SP projection
    for i=1:(2L+1) **do**
      $\hat{\mathcal{X}}_{k,i} = \tilde{\mathcal{X}}_i^+ + \int_{\tilde{t}}^{t_k} \boldsymbol{f}^+(\tilde{\mathcal{X}}_i^+, \mathcal{X}^w_{k-1,i})dt$
    end for
  end if

  % Calculate mean and covariance
  $\hat{\mathbf{x}}_k = \sum_{i=1}^{2L+1} \mathbf{W}_i^m \hat{\mathcal{X}}_{k,i}$
  $\hat{\mathbf{P}}_{xx} = \sum_{i=1}^{2L+1} \mathbf{W}_i^c [\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}][\hat{\mathcal{X}}_{k,i} - \hat{\mathbf{x}}]^T$
end if

state $\boldsymbol{x}_0 = [-0.0695\ -0.0980\ -0.3205\ -0.1930]^T$. This value was used as initial state for all trials, as only 0.9 sec. of further integration would result in the deterministic system reaching the switching boundary. In practice, all simulations were further integrated by 3 sec. to ensure the switching boundary was met.

Similarly, initial estimate covariance for all trials $\hat{\boldsymbol{P}}_0$ was calculated by using the UKF to estimate the state of the deterministic system as it evolved from the fixed point to a time of 3 sec. This instance of the filter had initial process covariance and measurement covariance of $1e^{-3}\boldsymbol{I}_n$, where $\boldsymbol{I}_n$ is the identity matrix of size n. Noiseless measurements were performed on the deterministic system every 0.1 sec.

The continuous system dynamics is given by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{f}^\star(\boldsymbol{x}_k, \boldsymbol{w}_k)dt$ (see Eqns. 1 to 5 for more details). For the simplest walker, the dimension of the state space is $n_x = 4$. We assume full state observation, $\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{v}_k$ and hence $n_y = 4$. The simulated process and measurement noise are zero mean, normally distributed, with standard deviations σ_w and σ_v, respectively. The filters have assumed process and measurement noise which are zero mean, normally distributed, with standard deviations $\bar{\sigma}_w$ and $\bar{\sigma}_v$, respectively. The covariance matrices for the filter are $\boldsymbol{Q} = \bar{\sigma}_w^2 \boldsymbol{I}_{n_x}$ and $\boldsymbol{R} = \bar{\sigma}_v^2 \boldsymbol{I}_{n_y}$. Filter parameters, α, β, κ, were set to $1e^{-3}$, 2, and 0, respectively, which are typical per Wan and Van Der Merwe (2000).

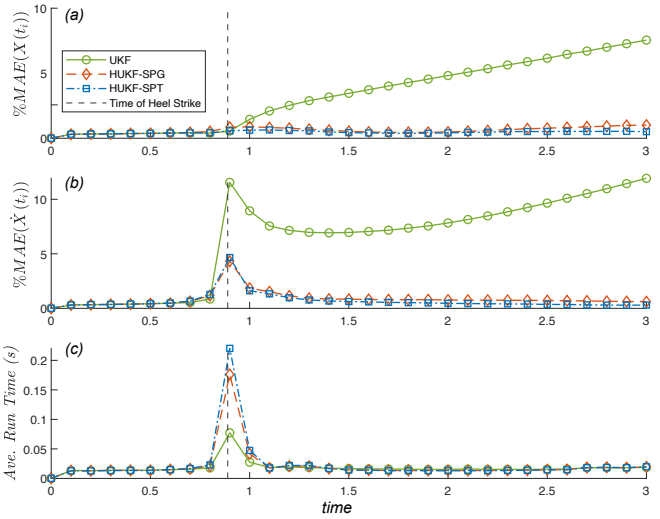

Fig. 3. Simulation results: (a) Mean Squared Error for the position, (b) Mean Squared Error for velocity, (c) Average run time as a function of time.

The study here is based on 6400 simulations. Combinations of the following values for process and measurement noise were chosen; process noise standard deviation for simulation $\sigma_w \in [0.0\ 0.001\ 0.01\ 0.1]$, estimated process noise standard deviation for filter $\bar{\sigma}_w \in [0.0001\ 0.001\ 0.01\ 0.1]$, and the measurement noise for simulation and filter were assumed to be identical $\sigma_v = \bar{\sigma}_v \in [0.0001\ 0.001\ 0.01\ 0.1]$. These three standard deviations gave 4^3 combinations. 100 trials were simulated for each combination, resulting in a total of $4^3 \times 100 = 6400$.

Computations were done in MATLAB 2020b on Windows 10 operating system. The processor was an Intel Core i5-2500K CPU with speed of 3.30GHz, 4 cores, and 16 GB of RAM. All integrations were done using *ode113* with an absolute tolerance of 10^{-13} and relative tolerance of 10^{-12}. The *ode113* has an in-built root finder for which can be accessed using the 'events' parameters. The root finder *fsolve* was used with a function tolerance of 10^{-12} was used to detect means meeting the switching boundary in Tables 3 and 4.

Figs. 3 (a) and (b) shows the mean absolute error (MAE) as a function of time for the angles and the angular rates, respectively, for the successful trials. These errors are computed as follows

$$MAE(X) = \frac{\sum_i^N |\theta_i - \hat{\theta}_i| + |\phi_i - \hat{\phi}_i|}{2N}$$
$$MAE(\dot{X}) = \frac{\sum_i^N |\dot{\theta}_i - \hat{\dot{\theta}}_i| + |\dot{\phi}_i - \hat{\dot{\phi}}_i|}{2N}$$

where the difference is taken between the actual and the estimate value and $N = 5269$ is the number of successful trials (see Sec. 4 for more details on trials which were excluded). The individual values of angle and angular rates were normalized against the approximate maximum value in the simulation, which is 0.4 for both the angles and angular rates. These plots demonstrate that all three algorithms have similar errors untill they reach the switching boundary. This is because all algorithms are identical up to this point. Thereafter, the UKF shows a sharp increase in error that appears to diverge. HUKF-SPG and HUKF-SPT show similar errors with the HUKF-SPT performing marginally better.

Fig. 3 (c) shows the computational time for the three filters as a function of iteration over the $N = 5269$ trials. All filters have the same computational time away from the switching boundary, as expected. At the switching boundary, UKF is the least computationally intensive, followed by HUKF-SPG and HUKF-SPT. Note that the absolute numbers are dependent on the computational capabilities of the system, the integrator, and the tolerances for the integration and the root finder. In comparing their relative magnitudes, it can be seen that HUKF-SPG and HUKF-SPT need 2.33 and 3 times more computations compared to HUKF, respectively.

4. DISCUSSION AND CONCLUSION

The paper presented two novel hybrid extensions of the UKF. The two extensions are identical to the UKF away from the switching boundary but use different time updates near boundary. The standard UKF propagates the SPs through the hybrid dynamics, allowing each sigma point to reach the switching boundary at different instants of time. The first extension, HUKF-SPG, *generates new SPs* at the switching boundary once the mean of the propagated SPs is on the switching boundary. The second extension, HUKF-SPT, *transforms the SPs* through the switching boundary based on the time that the mean reaches the switching boundary.

We chose to discard 49 or $\sim 1\%$ of the 6400 trials because the UKF covariance matrix $\hat{\mathbf{P}}$ ceased to be positive definite. The exact cause of this not yet known. We suspect this occurs when only some of the SPs transition through the switching boundary, as shown in Fig. 1 a2. In this condition, the SPs no longer represent a normal distribution, but a mixture model of two truncated normal distributions.

We chose to discard 1082 or $\sim 17\%$ out of 6400 trials because the measurement update (Table 2, line 13) caused a hybrid transition, i.e. $S(\hat{\boldsymbol{x}}_k) < 0$ and $S(\hat{\hat{\boldsymbol{x}}}_k) \geq 0$. Our current work does not consider hybrid dynamics in the measurement update, which we believe requires special treatment, and is considered the focus of future work.

The UT projects SPs to estimate a normal distribution whose mean is the estimated state. However, it is known that the true state is not normally distributed when near the switching boundary and, as a result, the UT becomes less accurate. Since the UKF projects SPs via the UT through both domains $V^{\pm}$, and the switching boundary Σ, but still assumes normal distribution, it results in the highest MAE. HUKF-SPG and HUKF-SPT do not expose SPs to both $V^{\pm}$ and Σ when projecting them with the UT, and thus produce less error. A caveat of HUKF-SPT is that it requires backward integration of the dynamics as shown in Fig. 1 c6. This is possible for the simplest walker because it is a conservative system. However, the backward integration is not possible for non-conservative system. The HUKF-SPT and HUKF-SPG require more computational time than UKF due to need for additional root finding.

We conclude, that of the tested estimators, both novel extensions outperform the standard UKF at the cost of moderately more computational time, and the results justify further investigation. The drawback of the more accurate HUKF-SPT is that it only works for conservative systems as it requires backward time integration.

REFERENCES

Bhounsule, P.A. (2014). Numerical accuracy of two benchmark models of walking: the rimless spoked wheel and the simplest walker. *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms*, 137–148.

Bloesch, M., Hutter, M., Hoepflinger, M.A., Leutenegger, S., Gehring, C., Remy, C.D., and Siegwart, R. (2013). State estimation for legged robots-consistent fusion of leg kinematics and imu. *Robotics*, 17, 17–24.

Gao, Y., Yuan, C., and Gu, Y. (2021). Invariant extended kalman filtering for hybrid models of bipedal robot walking. *IFAC-PapersOnLine*, 54(20), 290–297.

Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M. (1998). The simplest walking model: stability, complexity, and scaling. *Journal of Biomechanical Engineering*, 120(2), 281–288.

Grizzle, J., Choi, J.H., Hammouri, H., and Morris, B. (2007). On observer-based feedback stabilization of periodic orbits in bipedal locomotion. *Proc. Methods and Models in Automation and Robotics*, 27–30.

Hartley, R., Jadidi, M.G., Grizzle, J.W., and Eustice, R.M. (2018). Contact-aided invariant extended kalman filtering for legged robot state estimation. *arXiv preprint arXiv:1805.10410*.

Julier, S., Uhlmann, J., and Durrant-Whyte, H.F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, 45(3), 477–482.

Julier, S.J. and Uhlmann, J.K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), 401–422.

Kong, N.J., Payne, J.J., Council, G., and Johnson, A.M. (2021). The salted kalman filter: Kalman filtering on hybrid dynamical systems. *Automatica*, 131, 109752.

Raibert, M. (2010). Dynamic legged robots for rough terrain. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 1–1. IEEE.

Strogatz, S.H. (2018). Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering.

Wan, E.A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, 153–158. Ieee.