

# A ROBUST FRAMEWORK FOR FAILURE DETECTION AND RECOVERY FOR TERRAIN-RELATIVE NAVIGATION

Sarah E. Houts\*  
shouts@stanford.edu

Shandor G. Dektor\*  
sgd@stanford.edu

Stephen M. Rock\*<sup>†</sup>  
rock@stanford.edu

\*Aerospace Robotics Laboratory, Stanford University  
496 Lomita Mall, Rm 250, Stanford, CA 94305, USA

<sup>†</sup>Monterey Bay Aquarium Research Institute (MBARI)  
7700 Sandholdt Road, Moss Landing, CA 95039, USA

**Abstract**—Terrain-Relative Navigation (TRN) is an emerging technology used for localization of autonomous underwater vehicles (AUVs). TRN provides a map-relative position estimate for vehicle navigation, and has been demonstrated with meter-level accuracy. However, in adverse conditions, such as operating over areas of terrain change (i.e. mass transport events) or artifacts of the map-generation process, a TRN filter can become lost in the incorrect map, yielding bad position estimates. These conditions are aggravated by the combination of high precision inertial navigation systems and particle filters. To alleviate these situations of incorrect position estimates, a robust TRN framework has been designed that triggers a reinitialization when errors are detected to recover the correct map-relative position.

This paper focuses on an adaptation of the Normalized Innovations Squared (NIS) metric commonly used to detect failures in Extended Kalman Filters (EKF). The NIS metric measures whether or not the vehicle measurements are consistent with the expected measurements relative to terrain variation and sensor noise. When the NIS metric exceeds the calculated threshold, a failure is declared and a filter reinitialization is triggered – expanding to a broad area search in order to converge to the correct position estimate. Results are shown of the robust framework successfully reinitializing using field data from MBARI *Dorado*-class AUV runs in Monterey Bay.

## I. INTRODUCTION

Terrain-Relative Navigation (TRN) provides a map-relative position estimate and has been demonstrated with meter-level accuracy by a number of different groups [1], [2], [3]. With greater use, TRN algorithms are operating in a wider variety of situations, going beyond ideal conditions. Such adverse conditions include operating in areas where the terrain has changed (i.e. mass transport events) or in areas of the map where there are artifacts of the map-generation process. In these adverse conditions, a TRN filter can become locked on an incorrect position estimate, and this can lead to failures, for example colliding with the terrain.

This issue is aggravated by using high precision inertial navigation systems (INS) with Particle Filters (PFs). PFs are frequently used in TRN applications because of their ability to handle non-linearities and to estimate many different variables, including orientations, sensor biases, or terrain motion. However, when combined with the low process noise appropriate for a high precision INS, they can fall victim to particle deprivation as they trade state space coverage for precision.



Figure 1: A *Dorado*-class AUV at MBARI. Photo credit: Peter Kimball.

To demonstrate the need for failure detection and recovery in a TRN system, consider the example shown in Figures 2 and 3. In this example, a map artifact has been simulated by shifting the left half of a self-consistent map south by 30 meters.

The autonomous underwater vehicle (AUV) is traveling from right to left across the map. In Figures 2a and 3a, the blue dashed line shows the true map-relative position. The green and red markers indicate a converged TRN estimate, with a filter covariance below a threshold of  $64 \text{ m}^2$ . Above this threshold the estimate is considered unconverged and is not used by the vehicle control. The green markers indicate that the TRN estimate is correct, while the red indicate that it is incorrect. In Figures 2b and 3b, the position estimate bias with respect to the true map-relative position is shown along with the TRN filter covariance.

Figure 2 uses an existing TRN filter presented in previous work by the authors [4]. This filter works well in nominal conditions, and converges to the correct map-relative estimate before reaching the discontinuity in the map. When it reaches the discontinuity just before 900 seconds, it fails to capture the shift, and remains tightly converged on an incorrect estimate.

Figure 3 uses the same filter, but with the addition of a

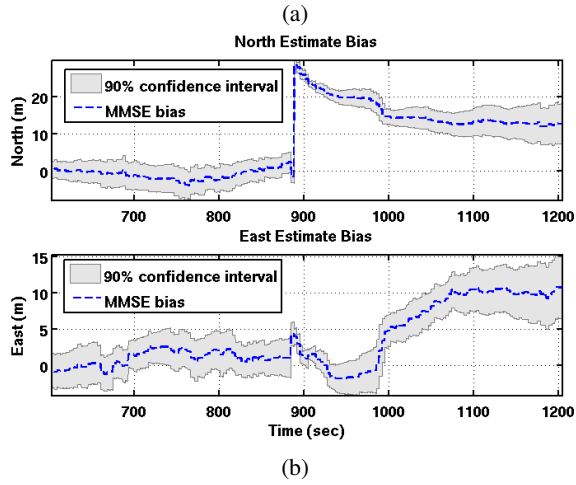
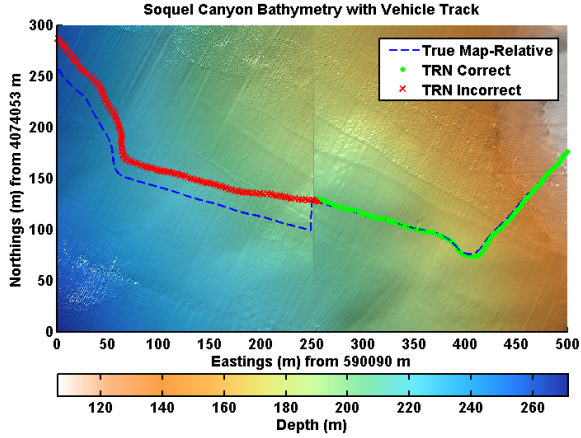


Figure 2: TRN filter progressing across simulated map artifact without the robust TRN framework. (a) TRN estimate and true map-relative trajectory for an artificial map artifact. The vehicle travels from right to left. The filter does not identify the shift in map across the artifact. The blue dashed line shows the true map-relative position of the vehicle. Green and red markers indicate a converged TRN estimate, correct and incorrect respectively. (b) Mean with respect to truth and 90% confidence bounds for the North and East position estimates.

robust TRN framework which performs diagnostic checks. These checks identify that the estimate is incorrect after crossing the discontinuity and trigger filter reinitializations, shown in Figure 3a as black stars, eventually converging on the correct position estimate. These reinitializations can be seen in Figure 3b as large increases in the covariance of the estimate.

This paper details the robust framework used in the above example. The framework and reinitialization process are discussed, along with a detailed explanation of the triggers, focusing on the Normalized Innovations Squared (NIS) metric that is demonstrated above.

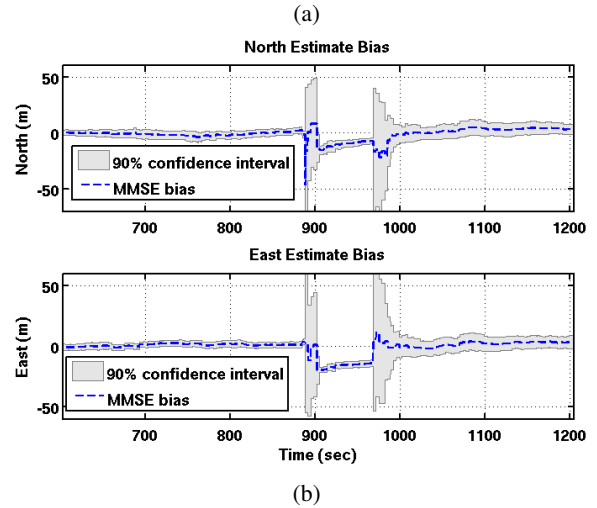
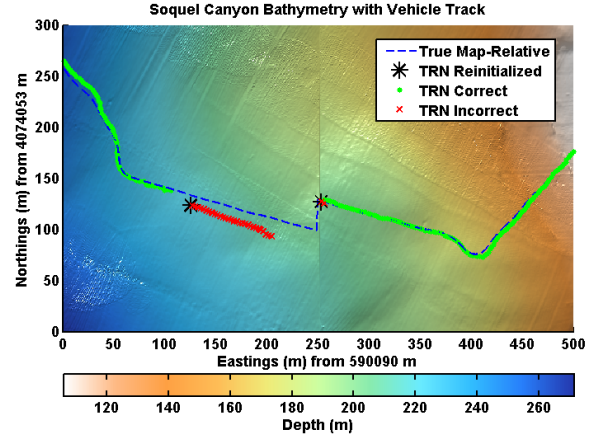


Figure 3: TRN filter progressing across simulated map artifact with the robust TRN framework active. (a) TRN estimate and true map-relative trajectory for an artificial map artifact. The vehicle travels from right to left. Green and red markers indicate a converged TRN estimate, correct and incorrect respectively. Filter reinitialization occurs at the points indicated with black stars as the failure is identified. (b) Mean with respect to truth and 90% confidence bounds for the North and East position estimates.

## II. RELATED WORK

Several techniques exist for recognizing filter failure and responding to it, both in general and TRN-specific. In general, these tests look at the likelihood of measurements given the estimated vehicle state. The checks introduced by [5] evaluate whether the normalized innovations of a Kalman Filter are white, uncorrelated, and with unit standard deviation. A variant of the method in [5], the Normalized Innovations Squared (NIS) test, is outlined for Kalman Filters in [6]. This method is focused on whether the filter innovations follow the expected unit standard deviation. This NIS metric is discussed in more detail in Section IV-B.

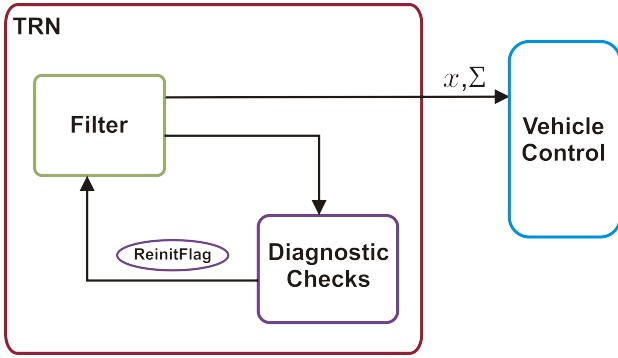


Figure 4: Depiction of the robust TRN framework. Diagnostic checks are performed on the filter, triggering reinitializations when necessary.

Tests for non-parametric filters have also been developed. For example, the augmented Monte-Carlo localization (AMCL) algorithm developed in [7] compares the ratio of the near-term and longer-term measurement likelihood; rapid changes in the measurement likelihood are assumed to be caused by filter failure. The NIS test and  $\chi^2$  tests are adapted in [8] and [9], respectively, for use with particle filters.

Integrity checks have been used in some TRN applications. An NIS check was used in [10] to monitor the health of an Unscented Kalman Filter (UKF) estimating the position of an automobile along a track using slope measurements. In another example, the TerrP system used in [11] checks the integrity of an AUV DVL-based TRN Particle Filter position estimate using a set of integrity check heuristics based on the suitability of the terrain and the goodness of fit.

Designing the response to filter failure is dependent on the application. One response to failures in Kalman Filters is to increase the process noise [6], typically used for tracking applications with unknown inputs. The AMCL algorithm in [7] used the ratio of short-term and long-term measurement likelihoods to determine the number of random particles injected in a robo-soccer playing field to account for manual robot re-positioning. The innovations monitor in [10] triggered a transition from a UKF to a Particle Filter, as it assumed that the non-linearities in the system caused system failure.

This paper adapts the NIS test from [8] to evaluate failure of non-parametric TRN filters. This test was chosen because it is fast to compute and is unlikely to have false triggers for TRN applications. Techniques such as AMCL can be falsely triggered by changes in terrain information content, while the NIS metric accounts for that. The response selected in this framework is to reinitialize the filter over a broad area when a filter diagnostic check is triggered.

### III. ROBUST TRN FRAMEWORK

The implementation of the robust TRN framework presented here is based on the structure depicted in Figure 4. The TRN filter passes a position estimate and covariance to the vehicle control system. Meanwhile, diagnostic checks monitor information from the filter to ensure that the estimate is good.

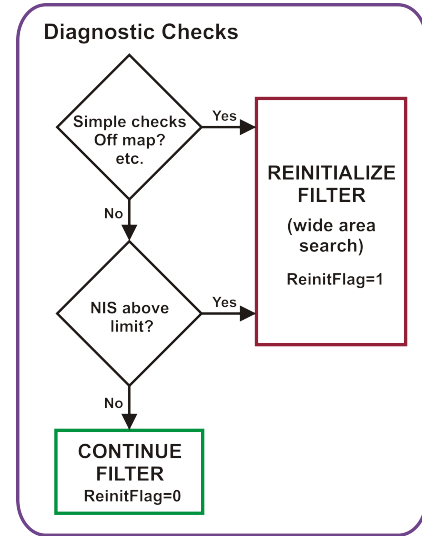


Figure 5: Depiction of the diagnostic checks performed by the robust TRN framework. Simple checks such as if the estimate is off the map are performed, and the windowed average NIS is compared to a threshold. If either of these trigger a reinitialization, the filter is reinitialized to a wide area search.

The two categories of diagnostic checks are shown in Figure 5, and consist of simple heuristics such as the time since the last measurement update, as well as the more detailed filter consistency check on the innovations of the filter. These are described in more detail in Section IV.

When a diagnostic check is failed, this system triggers a reinitialization of the TRN filter. The key aspect of the reinitialization is to expand the search region to a broad area to recapture the correct position. This is particularly important for use with PFs, which may have fallen victim to particle deprivation.

### IV. FILTER DIAGNOSTIC CHECKS

The filter diagnostic checks fall into two categories: checking on the update regularity (i.e. has the filter been able to do motion and measurement updates?), and the estimate consistency (i.e. does the estimate agree with what the vehicle is observing?). These diagnostic checks inform the filter when a problem has arisen, so that it may take the appropriate step of reinitializing the filter.

#### A. Filter Update Regularity

This category of diagnostic checks is straightforward. The goal is to identify situations where the filter may have wandered away from a good estimate during a period of outages that have prevented regular motion or measurement updates. These are situations that deviate from the nominal operation of the TRN filter and may cause the filter not to capture the correct position.

Causes for being unable to perform measurement updates include not having range measurements and not having a map to correlate the range measurements against. Lack of motion

updates can occur when there are no velocity measurements from the Doppler Velocity Logger (DVL).

In this framework, these failure modes are handled using a simple heuristic limit on the time since the last good update. If it has been too long since an update occurred, a reinitialization is triggered.

### B. Filter Consistency

The goal of filter consistency checks is to identify when the filter has an incorrect estimate. An incorrect estimate may be caused by a variety of issues, for example terrain change or artifacts of the map-generation process. The validity of an estimate can be evaluated based on whether the measurements predicted by the estimate agree with the observed measurements. This is typically done by monitoring the innovations or the error residual of the filter. The consistency check assesses whether the measured innovations follow their expected statistical properties.

The innovations check used in this paper is the Normalized Innovations Squared (NIS) test. Specifically, a flag is thrown if the windowed average of the NIS metric exceeds a threshold. This test is chosen due to its robustness and ease of implementation. The method of calculating the NIS, the windowed average, and determining the NIS threshold are detailed below.

1) *NIS for Kalman Filters:* The NIS check for Kalman Filters is based on the assumption that, under nominal conditions, the normalized innovations of a Kalman Filter will have unit variance. The innovations of a Kalman Filter are the difference between the actual measurement and the expected measurement.

$$\tilde{y}_k = y_k - \hat{y}_{k|k-1} \quad (1)$$

Assuming Gaussian noise on the measurement, the innovations are zero mean with a covariance  $S_k$ , where

$$S_k = R_k + H_k P_{k|k-1} H_k^T. \quad (2)$$

$H_k$  is the measurement linearization,  $P_{k|k-1}$  is the predicted state covariance, and  $R$  is the measurement covariance. The normalized innovation is then defined as

$$\eta_k = S_k^{-\frac{1}{2}} \tilde{y}_k \quad (3)$$

and the Normalized Innovations Squared is

$$\begin{aligned} NIS_k &= \tilde{y}_k^T S_k^{-1} \tilde{y}_k \\ &= \eta_k^T \eta_k. \end{aligned} \quad (4)$$

As  $\eta_k$  is expected to be a zero-mean Gaussian with unit variance, the NIS is expected to follow a  $\chi_n^2$  distribution where  $n$  is the number of measurements. The NIS therefore serves as a measure of the likelihood of getting the innovation  $\tilde{y}_k$ . If the innovations are large relative to their expected standard deviation, it means that the expected measurements are not in good agreement with the actual measurement.

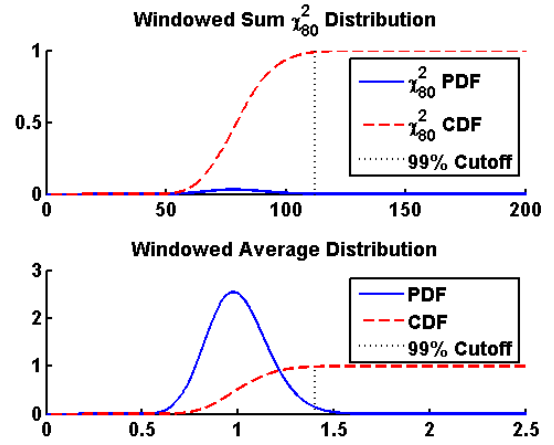


Figure 6: The  $\chi^2$  NIS distribution for a window of 20 measurements. The upper plot and lower plots show the PDF and CDF for the windowed sum and windowed average, respectively. The 99 percent threshold of the CDF is shown with a dotted line, which corresponds to a threshold of 1.4.

2) *Windowed NIS Failure Detection:* Robust failure detection is implemented by monitoring a windowed average of the last  $l$  NIS measurements. That is,

$$\overline{NIS}_k = \frac{1}{l} \sum_{j=k-l}^k NIS_j. \quad (5)$$

Under nominal conditions, the windowed sum follows a  $\chi_{ln}^2$  distribution. Figure 6 shows the expected windowed sum and windowed average NIS distribution. This is shown for a configuration using four DVL beams with a 20 measurement length window. The NIS average threshold can be chosen from this windowed average CDF.

The choice of both window length and NIS threshold involve failure detection performance tradeoffs. Increasing the window length makes the filter diagnostic less sensitive to spurious measurements, but increases the time required to identify a failure. The choice for the value of the NIS threshold trades between false positives and false negatives.

The filter diagnostic check here applies a 99 percent threshold for reliability – if the filter estimate is correct, the NIS threshold will only be triggered 1 percent of the time. In Figure 6, for a window length of 20, this corresponds to a value of 1.4.

3) *NIS adaptation to non-parametric TRN filters:* The NIS method can be adapted to non-parametric filters in a straightforward manner. The innovation  $\tilde{y}_k$  and its covariance matrix  $S_k$  are derived from approximating the distribution of expected measurements as gaussian. A non-parametric filter represents its state as

$$\mathcal{X}_k := x_k^{[1]}, x_k^{[2]}, \dots, x_k^{[M]} \quad (6)$$

where each sample  $x_k^m$  ( $1 < m < M$ ), is a discrete hypothesis with its own likelihood  $w_k^{[m]}$  and expected measurement  $\hat{y}_k^{[m]}$ ,

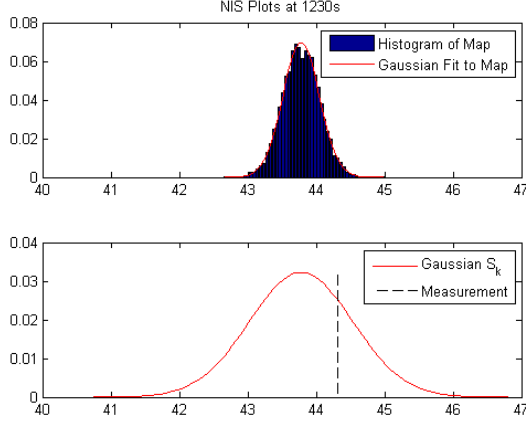


Figure 7: The upper plot shows the contribution of map variance to  $S_k$ .  $S_k$  and the measurement  $y_k$  are shown in the lower plot.

where

$$\hat{y}_k^{[m]} = [\hat{y}_{1,k}^{[m]} \hat{y}_{2,k}^{[m]} \dots \hat{y}_{n,k}^{[m]}]^T \quad (7)$$

$$\hat{y}_{i,k}^{[m]} = f(x_k^{[m]}, \hat{h})_i \quad (8)$$

where  $n$  is the number of range measurements correlated, and the estimate  $\hat{y}_{i,k}^{[m]}$  of beam  $i$  is a function of the vehicle position  $x_k^{[m]}$  and the map  $\hat{h}$ . The quantities  $\tilde{y}_k$  and  $S_k$  are then computed as follows:

$$\mu_k = \sum_{m=1}^M w_k^{[m]} \hat{y}_k^{[m]} \quad (9)$$

$$\tilde{y}_k = y_k - \mu_k \quad (10)$$

$$S_k = R_k + \sum_{m=1}^M w_k^{[m]} (\hat{y}_k^{[m]} - \mu_k) (\hat{y}_k^{[m]} - \mu_k)^T \quad (11)$$

The normalized innovations are then calculated by substituting the  $S_k$  and  $\tilde{y}_k$  calculated for the non-parametric filter into Equation (4). An example of this process for calculating  $S_k$  and  $\tilde{y}_k$  is illustrated in Figure 7.

Although the NIS metric may not be exact for the often non-linear nature of particle filter measurement updates, it takes into account the impact of position uncertainty and terrain variation, providing a useful metric for identifying incorrect filter estimates.

## V. RESULTS

The results presented in Section I used a simulated map artifact to demonstrate the utility of the diagnostic checks and reinitialization to detect the failure and recover the correct position estimate.

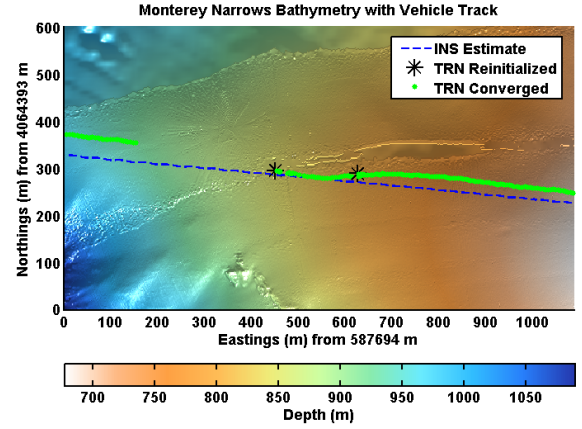
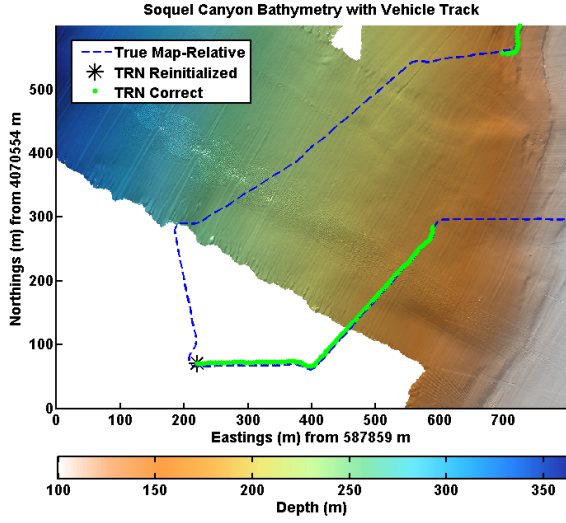
In this section, two cases are presented using field data from MBARI's AUVs, exhibiting the use of both categories of diagnostic check. First, an online field trial was performed with

the vehicle leaving the mapped area for an extended period of time. This demonstrates the trigger based on the time since the last measurement update. Second, field data from a run that went over a map artifact is shown demonstrating the NIS trigger for reinitialization.

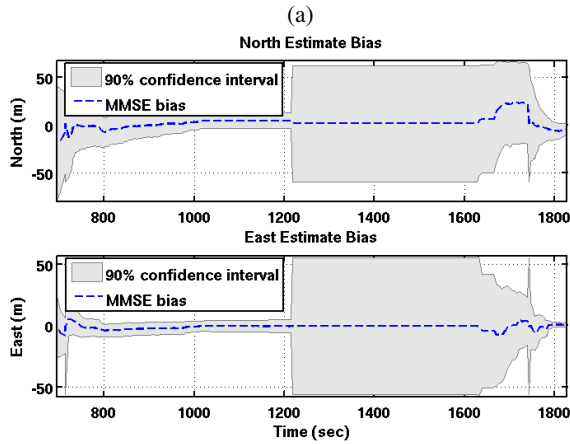
A demonstration of the robust TRN framework triggering a reinitialization from a simple check is shown in Figure 8. This is a field trial in which the vehicle left the mapped area in Soquel Canyon in Monterey Bay, and all computations were done in real time on-board the vehicle. Figure 8a shows the vehicle trajectory displayed over the map, as it proceeds clockwise. Here, the blue dashed line shows the true map-relative position, which is calculated using a technique that utilizes many TRN trajectories over the same map to find the best fit for any geo-registration errors. The green markers show the converged TRN position estimate, with a covariance below a given threshold of  $64 \text{ m}^2$ . The reinitialization of the filter is shown by the black star. In Figure 8b the offset between the TRN estimate and the true position is shown along with the filter covariance. The expansion to a wide area search can easily be seen in the abrupt jump in covariance just after 1200 seconds. Once the vehicle has returned to the mapped area and is able to compute measurement updates, the filter once again converges to the correct position. For this initial demonstration, a long delay was added between when the vehicle identified that it had left the map and when the filter was reinitialized. The reinitialization took place as expected.

A demonstration of the NIS metric triggering filter reinitializations is shown in Figure 9. This shows field data from a run near the Narrows in Monterey Bay where the vehicle crossed an artifact of the map-generation process as it traveled from right to left across the map. This map artifact was unanticipated, and as a result, the true map-relative position is unavailable. As before, the green markers show the TRN estimate with filter covariance less than the threshold of  $64 \text{ m}^2$ . However, since the true map-relative position is not available for this run, the estimate from the INS is shown as the blue dotted line. The reinitializations are shown with black stars. In Figure 9a, the reinitialization can be seen as the vehicle approaches the map artifact. After crossing it, the filter converges to a clearly incorrect position, which it identifies, triggering a second reinitialization. After this reinitialization, the filter converges once again. The filter estimate relative to the INS and covariance are shown in Figure 9b. The increase in covariance when the filter reinitializes can be clearly seen around 3200 seconds and 3400 seconds. Figure 9c shows the value of the NIS metric during this run. The value for each four-beam DVL measurement is shown in blue, with the windowed average in red. The threshold for triggering a reinitialization is 1.4, shown as the green dashed line. As soon as the windowed average NIS value reaches the threshold, a reinitialization of the filter is triggered.

These runs demonstrate the utility of the robust TRN framework, and the NIS metric in particular, allowing the filter to identify incorrect estimates and reinitialize to reacquire the correct position.



(a)

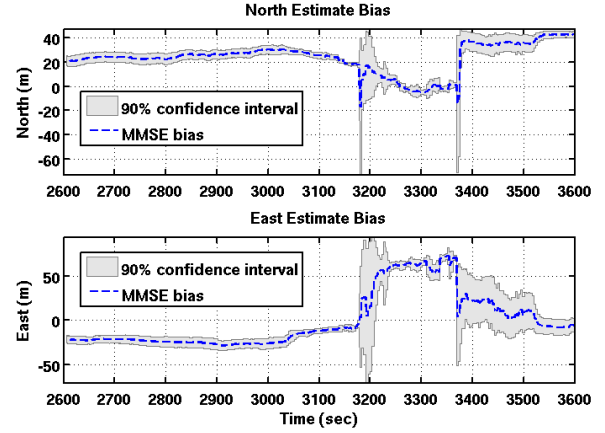


(b)

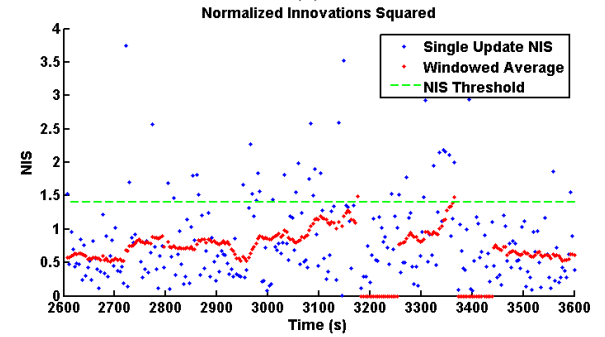
Figure 8: TRN filter leaving the mapped area with the robust TRN framework active. (a) TRN estimate and true map-relative trajectory. The vehicle travels clockwise. Reinitialization occurs at the black star. After returning to the map, the filter converges to the correct position. (b) Mean with respect to truth and 90% confidence bounds for the North and East position estimates. The vehicle leaves the mapped area around 1000 seconds.

#### ACKNOWLEDGEMENTS

The authors would like to thank MBARI and NASA ASTEP NNX11AR62G for support, as well as the Stanford Graduate Fellowship and National Defense Science and Engineering Graduate Fellowship for partial funding of this work. Researchers at MBARI who were particularly helpful in acquiring the presented data include Hans Thomas, Rob McEwen, Brett Hobson, and Rich Henthorn.



(b)



(c)

Figure 9: A map artifact is detected with the robust TRN framework. The NIS metric triggers reinitialization twice. (a) The vehicle travels from right to left across an artifact in the map. The TRN filter detects an error and reinitializes the filter twice at the black stars. The blue line indicates the INS estimate. (b) Mean with respect to INS and 90% confidence bounds for the North and East position estimates. (c) The NIS metric. The windowed average of the NIS triggers reinitialization when it reaches the threshold.

## REFERENCES

- [1] I. Nygren, "Robust and efficient terrain navigation of underwater vehicles," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pp. 923 – 932, may 2008.
- [2] K. Anonsen and O. Hagen, "An analysis of real-time terrain aided navigation results from a HUGIN AUV," in *OCEANS 2010*, pp. 1 – 9, Sept. 2010.
- [3] S. Dektor and S. Rock, "Improving robustness of terrain-relative navigation for AUVs in regions with flat terrain," in *Autonomous Underwater Vehicles (AUV), 2012 IEEE/OES*, pp. 1 – 7, September 2012.
- [4] D. Meduna, S. M. Rock, and R. McEwen, "AUV terrain relative navigation using coarse maps," in *Proceedings of the 2009 Unmanned Untethered Submersible Technology Conference* (R. Blidberg, ed.), (Durham, NH), August 2009.
- [5] R. Mehra and J. Peschon, "An innovations approach to fault detection and diagnosis in dynamic systems," *Automatica*, vol. 7, no. 5, pp. 637 – 640, 1971.
- [6] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2004.
- [7] J.-S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 454 – 459 vol.1, 2002.
- [8] M. Scalzo, G. Horvath, E. Jones, A. Bubalo, and M. Alford, "Adaptive filtering for single target tracking," *SPIE*, vol. 7336, 2009.
- [9] F. van der Heijden, "Consistency checks for particle filters," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 140 – 145, jan. 2006.
- [10] A. Dean, J. Langelaan, and S. Brennan, "Improvements in terrain-based road vehicle localization by initializing an unscented kalman filter using particle filters," in *American Control Conference (ACC), 2010*, pp. 700 – 707, July 2010.
- [11] O. Hagen, K. Anonsen, and M. Mandt, "The HUGIN real-time terrain navigation system," in *OCEANS 2010*, pp. 1 – 7, Sept. 2010.