

# Final Project

Jaden Wang

## 1 Introduction

Throughout the paper, we refer to the regular UKF as UKF and the manifold version of UKF as UKF-M. Moreover, to avoid terminology collision, we use the term “direction” instead of “orientation” to describe the body frame relative to the inertial frame and reserve the term “orientation” for its meaning in manifold theory, *i.e.* a global coherent choice of handedness for the tangent spaces of a manifold. Finally, all manifolds are assumed to be smooth.

As we have seen in class, UKF is accurate up to third order without sacrificing speed significantly for low-dimensional problems, making it an attractive alternative to EKF. Moreover, it does not suffer from the sample impoverishment problem of particle filters and saves lots of computational cost by choosing the particles judiciously. Therefore, UKF hits a sweet spot between computational complexity and accuracy for low-dimensional problems. However, in many engineering applications, especially in robotics and spacecraft control, we wish to estimate states that are confined to a manifold instead of the Euclidean space. For example, if a spacecraft can only perform translations and rotations, then its pose is an element of the manifold/Lie group  $SE(3)$ . The manifold can introduce high non-linearity to the system, potentially significantly debase the performance of filters designed for the Euclidean space. This motivates designing the filters natively for manifolds. However, this approach has several challenges. First, manifolds are much more difficult to understand and often require substantial pure math training, making them inaccessible to many practitioners. Second, most techniques that we take for granted in Euclidean space, such as statistics, do not generalize easily to manifolds. It takes lots of care to generalize a Euclidean filter to the manifold case. Third, we must restrict the class of manifolds in order to have any hope of successfully making a filter, but being too restrictive can significantly limit the applications. Therefore, it would be desirable to design the filter for a large class of manifolds that encompass the majority of manifolds we encounter in aerospace engineering problems, ideally relying only

on intuitive concepts from manifold theory.

In this paper, we present the theory and algorithm from []. We will see that by restricting to *parallelizable manifolds*, we can adapt UKF to the most common manifolds using concepts that can be intuitively understood.

## 2 Manifold Theory

Stripping away much jargon and abstraction, we provide an informal treatment of the necessary concepts from manifold theory so that readers with minimal background in the subject can still acquire sufficient intuition for the purpose of understanding this paper.

**Definition 2.1** — A smooth manifold  $M$  of dimension  $n$ , denoted  $M^n$ , is a smooth space that locally looks like a Euclidean space  $\mathbb{R}^n$ .

A rigorous definition requires the technicality of local charts and topological properties that are outside the scope of this paper. The following intuition should suffice:

**Example 2.2** (Earth)

Although the surface of the Earth is a sphere  $S^2$ , a 2-dimensional manifold, it locally looks flat and resembles  $\mathbb{R}^2$ .

**Example 2.3** ( $S^n$ )

All hyperspheres  $S^n$  are  $n$ -dimensional manifolds. Note that  $S^1$  is just a circle. It locally looks like a line which is a copy of  $\mathbb{R}$ . We use angles  $\theta \in (-\pi, \pi]$  to represent points on  $S^1$  in this paper.

**Definition 2.4** — A **diffeomorphism** from a smooth manifold  $M^n$  to another smooth manifold  $N^n$  is an invertible smooth map. If such a map exists, we say  $M$  is diffeomorphic to  $N$ , denoted  $M \cong N$ .

A diffeomorphism declares that the two manifolds have the same manifold structure and

only differ superficially in the naming labels.

### Example 2.5

We can identify the surface of the Earth with a unit sphere in  $\mathbb{R}^3$  by a diffeomorphism: we simply shrink each point on the surface toward the center of the Earth until its distance to the center is 1.

**Definition 2.6** — A **Lie group** is a group that is also a smooth manifold.

### Example 2.7 ( $SO(n)$ )

The group of orientation-preserving rotations in any dimension  $n$ , denoted  $SO(n)$ , consists of rotation matrices in  $n$ -dimension. That is, the matrix is orthogonal and has determinant 1. It is a group: matrices compose under matrix multiplications. It is also a smooth manifold (by the regular value theorem).

When  $n = 2$ , elements of  $SO(2)$  are rotation matrices that can be parametrized by rotating angles, which can be viewed as elements of the circle  $S^1$ . It is not hard to check that the map  $\text{Rot} : S^1 \rightarrow SO(2), \theta \mapsto \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$  is a diffeomorphism with inverse being  $\text{atan2}$  on the first column, so  $S^1 \cong SO(2)$ . This implies that  $SO(2)$  is a 1-dimensional manifold.

We note that  $S^1$  is also a Lie group: its group operation is addition of angles. In fact,  $\text{Rot}$  is a Lie group isomorphism, respecting the group structure:  $\text{Rot}(\theta_1 + \theta_2) = \text{Rot}(\theta_1) \text{Rot}(\theta_2)$ . Thus, multiplication in  $SO(2)$  is commutative.

For simplicity and intuition, let us take the extrinsic viewpoint that a manifold  $M^n$  is always embedded in a Euclidean space of potentially higher dimension (*e.g.* the Earth surface  $S^2$  is embedded in  $\mathbb{R}^3$ ). Then we have the following definition:

**Definition 2.8** — Let  $M^n$  be a smooth manifold. For any point  $x \in M$ , the **tangent space of  $M$  at  $x$** , denoted  $T_x M$ , is the  $n$ -dimensional hyperplane tangent to  $M$  at  $x$ .

We can think of tangent spaces as attaching a tangent copy of  $\mathbb{R}^n$  at each point. Tangent

spaces are easy to visualize for the sphere. TODOFIGURE

In other words, the tangent space  $T_x M$  provides the best linear (read: Euclidean) approximation of a neighborhood of  $M$  around  $x$  and thus provides a natural setting when we want to generalize results from Euclidean spaces to manifolds.

**Definition 2.9** — A **smooth vector field**  $V$  on a manifold  $M^n$  is a smooth function that assigns every point  $x \in M$  a tangent vector in  $T_x M$ .

Smooth vector fields are fairly intuitive: for each point on the manifold, we pick a vector from the tangent space of this point. For example, we can visualize a vector field on a sphere as a “hairy ball”: there is a tangent arrow coming out of each point on the sphere. We also want the vector field to vary smoothly as we move smoothly around the base points of these vectors. For example, the velocity of clouds at each coordinate on Earth give us a smooth vector field: as we move around the coordinates, the velocity vectors of the clouds at those coordinates varies smoothly.

**Definition 2.10** — A smooth manifold  $M^n$  is **parallelizable** if there exists a set of smooth vector fields  $\{V_1, \dots, V_n\}$  on the manifold such that for every point  $x \in M$ , the tangent vectors  $\{V_1(x), \dots, V_n(x)\}$  form a basis of the tangent space  $T_x M$  at  $x$ .

Since each vector field determines a (possibly zero) tangent direction for the tangent space of every point, if it is possible to find  $n$  vector fields that point in different (nonzero) directions for all points, then we get a basis of the tangent space at each point, thus making  $M^n$  parallelizable.

Parallelizable manifolds encompass a sizable class of manifolds useful for engineering, including all Lie groups. However, some common manifolds are excluded from this class. Let us see an example and a non-example.

**Example 2.11** ( $SO(2)$ )

To parallelize  $SO(2)$ , we only need to find a single smooth vector field. We can easily obtain a smooth vector field on the circle by just choosing 1 on each tangent line, *i.e.*

the positive unit vector in each copy of  $\mathbb{R}$ . Since  $S^1 \cong SO(2)$ , this corresponds to a smooth vector field on  $SO(2)$  via the diffeomorphism.

**Example 2.12** (non-example: the sphere  $S^2$ )

The hairy ball theorem famously says that you cannot comb a hairy ball without leaving behind a cowlick. That is, any smooth vector field on  $S^2$  must have at least a point where the vector over that point is zero. Thus we cannot get a basis at that point using only 2 vector fields.

It turns out that it is always possible to “lift” a much larger class of manifolds called *homogeneous spaces* to Lie groups, hence allowing the same algorithm to work on this class. The sphere, and most manifolds we care about in practice, are examples of such space. Due to time constraints, we are not going to address this generalization in the paper.

**Definition 2.13** — Let  $M$  be a manifold. A **geodesic** traces out a locally shortest path on  $M$  with constant speed.

This is intuitive: suppose you are walking on a non-flat terrain. You pick a direction and walk at constant speed in that direction. The path you trace out is a geodesic. Commercial planes fly along geodesics of the Earth.

**Definition 2.14** — The **exponential map of a compact manifold** is defined as follows: for any  $x \in M$ , take any tangent vector  $v \in T_x M$ , there exists a neighborhood  $U_x$  of  $x$  on  $M$  and a unique geodesic  $\gamma_v : [0, 1] \rightarrow U_x$  (guaranteed by ODE theory) that starts from  $x$  with tangent velocity  $v$ , and we set  $\exp_x(v) = \gamma_v(1)$  for each  $x$ .

Any tangent vector gives you the direction and speed for your walk. The exponential map tells you which point you are at after one unit time of walking. Note that we do not need compactness to define geodesics, but I add the assumption to simplify it. The manifolds of interest in this paper are compact.

**Proposition 2.15**

For every  $x \in M$ , the exponential map is a local diffeomorphism from some neighborhood  $O_x$  of the origin in  $T_x M$  to  $U_x$ .

Since the differential (read: Jacobian) of the exponential map at  $x$  is the identity (the tangent vector at  $x$  is precisely  $v$  itself), the proposition is an easy consequence of the inverse function theorem. As a local diffeomorphism, the exponential map is locally invertible. We expectedly call this local inverse the **logarithmic map**. At each base point  $x$ ,  $\log_x : U_x \rightarrow O_x$  takes a point  $y$  near  $x$  on the manifold and maps it to a vector in  $T_x M$ .

**Example 2.16 ( $SO(2)$ )**

Since  $SO(2)$  or  $S^1$  is compact, we can use the above definition for exponential map. Geodesics in  $S^1$  are arcs. Thus, the exponential map of  $S^1$  simply wraps a real number (from the tangent line) around the circle starting at the base point and outputs the end point of this wrapped arc. If the base point is  $\theta$  and the real number is  $v$ , then  $\exp_\theta(v) = \bar{v} + \theta$ , where  $\bar{v} = v \bmod 2\pi \in (-\pi, \pi]$ . By the diffeomorphism, exponential map of  $SO(2)$  outputs  $\text{Rot}(\bar{v}) \text{Rot}(\theta)$ . That is, for any base point  $C \in SO(2)$ , we have

$$\exp_C : T_C SO(2) \cong \mathbb{R} \rightarrow U_C, \quad v \mapsto \text{Rot}(\bar{v})C.$$

This will be quite handy for generating sigma points.

As for the logarithmic map, we have

$$\log_C : U_C \rightarrow O_C, \quad D \mapsto \text{atan2} \left( (DC^{-1})_{21}, (DC^{-1})_{11} \right).$$

We use `atan2` here since we want to have both positive and negative numbers for sigma points. But notice that we cannot recover the original real number  $v$  if it is outside  $(-\pi, \pi]$ ! That means the neighborhoods  $O_C = (-\pi, \pi]$  and  $U_C = SO(2)$  for any  $C \in SO(2)$ . Thus, although we have no restriction from  $U_C$ , we must not apply the exponential map to any number outside  $(-\pi, \pi]$  if we want to use its invertibility property. This is not a problem in practice since we take very small steps to generate sigma

points. Thus we can simply treat  $v \in (-\pi, \pi] \subseteq \mathbb{R}$  as an angle and forget about  $\bar{v}$ .

## 3 UKF-M

### 3.1 Considerations

In  $\mathbb{R}^n$ , we choose sigma points in a symmetric way so that we obtain nice cancellations of even-order terms in the Taylor expansion analysis and therefore propagate mean and covariance estimations up to third-order accuracy. The biggest hurdle to generalize this method to a manifold is that we cannot update the mean and covariance on the manifold as we do not even have addition defined. To overcome this hurdle, a natural idea is to first obtain the dispersion of sigma points on the manifold, map them to a common tangent space to update the statistical estimates, and then retract back to the manifold.

Recall that in the Euclidean case, we view the true state  $x$  as a Gaussian random variable with mean  $\hat{x}$  and covariance  $P$  coming from an zero-mean error term  $\varepsilon$ . That is,  $x = \hat{x} + \varepsilon$  with  $\varepsilon \sim (0, P)$ . In the manifold case, since  $M$  does not have a vector space structure, we cannot simply add error to the point. Thus we modify the assumption to

$$x = \phi(\hat{x}, \xi(\hat{x})), \quad \xi(\hat{x}) \sim \mathcal{N}(0, P) \quad (1)$$

where  $\xi$  is a vector field on  $M$  that encodes an error vector at each estimate  $\hat{x}$ , and  $\phi : M \times \mathbb{R}^n \rightarrow M$  is a user-defined smooth function that satisfies  $\phi(\hat{x}, 0) = \hat{x}$ . We also require the differential (read: Jacobian) of  $\phi$  at  $(\hat{x}, 0)$  to be the identity so that we can invert it locally. We call  $\phi$  the **retraction**. Note that for readers with manifold background, the retraction is in general a smooth function  $\phi : TM \rightarrow M$ . Since we are in the case of parallelizable manifolds, their tangent bundle  $TM \cong M \times \mathbb{R}^n$  is trivial.

We note that this formulation preserves the Euclidean nature of errors, allowing us to compute meaningful statistics. Indeed, here we view  $\xi(\hat{x})$  as a random vector in the tangent space at  $\hat{x}$ , and its components are coefficients under the basis given by the set of vector fields at  $\hat{x}$ . That is, we have  $\xi(\hat{x}) = \xi_1 V_1(\hat{x}) + \dots + \xi_n V_n(\hat{x})$ . Since the basis of  $\xi$  varies smoothly, we can smoothly propagate the covariance  $P$  forward using coordinates of  $\xi(\hat{x})$  at different  $\hat{x}$ . Without a global smoothly varying basis provided by the parallelizable condition, the  $P$

computed at previous iteration would have no meaning for the current iteration since it was computed using a completely different basis. An analogy is that in the Euclidean space, we suddenly decides to making our standard basis vectors 100 times bigger, so the coordinates are 100 times smaller. Under this basis change, the previous  $P$  now indicates 100 times more variance than it indicated one iteration prior, which is clearly spurious. Under a smoothly varying basis, we can meaningfully transport  $P$  to the new iteration. However, since  $\phi$  is most likely non-linear, we can no longer claim that the distribution of  $x$  is Gaussian even though the errors themselves are Gaussian.

The choice of the retraction map  $\phi$  gives more flexibility but potentially more headache. Luckily, there is a canonical retraction we can use for any smooth manifold: the exponential map that we encountered earlier. That is, we can set  $\phi(\hat{x}, \xi(\hat{x})) = \exp_{\hat{x}}(\xi(\hat{x}))$ . We allow a more general retraction map because the exponential map does not always have an explicit form for practical implementation.

The exponential map would allow us to sample  $2n$  sigma points at  $\hat{x}$  by flowing along  $n$  randomly sampled vectors  $\xi^{(i)}(\hat{x})$  forward and backward in time. In general, retraction map gives us the sigma points we need for implementing UFK-M.

### 3.2 The Algorithm

Now let us state the setup and assumptions of UKF-M. The system dynamics are described by the following equation

$$x_{k+1} = f(x_k, u_k, w_k), \quad (2)$$

where  $f : M \times \mathbb{R}^p \times \mathbb{R}^q \rightarrow M$  is a smooth function,  $u_k$  is a known input, and  $w_k \sim \mathcal{N}(0, Q_k)$  is a white noise. Note that the input and noise do not need to have the same dimension as the state, due to the generality of smooth function. The observations are obtained via

$$y_k = h(x_k) + v_k, \quad (3)$$

where  $h : M \rightarrow \mathbb{R}^m$  is smooth and  $v_k \sim \mathcal{N}(0, R_k)$ . Thus, the only modifications to the original setup are forcing the states to live on  $M$  and altering the domains of  $f$  and  $h$  accordingly.



To obtain the time update of the mean, we simply propagate  $\hat{x}_k$  forward assuming zero-noise:

$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k, 0) \quad (4)$$

Note that this is a departure from Simon’s implementation of UKF [], where he used the average of the propagated sigma points to compute  $\hat{x}_{k+1}^-$ . While it would be interesting to compare the two approaches in UKF, here we have no choice but to propagate the mean this way, since we can no longer take the average of points on a manifold. Although we can take the average once we map everything to the same tangent space, the tangent space needs to be based at some “average” point first. The only sensible choice is the  $\hat{x}_{k+1}^-$  from Equation 4.

Using the retraction map to sample sigma points  $\hat{x}_{k,i}$ , we propagate them through Equation 4 to obtain  $\hat{x}_{k+1,i}^-$ . In the Euclidean case, we can compute  $P_{k+1}^-$  from the propagated sigma points. But since these points are now on a manifold, we have to map them back to the same vector space to compare. The obvious candidate vector space is the tangent space of  $\hat{x}_{k+1}^-$ . Since we require the retraction map to be locally invertible between some neighborhoods  $O$  and  $U$ , we can use  $\phi^{-1}(\hat{x}_{k+1}^-, -) : U_{k+1} \rightarrow O_{k+1}$  to map all sigma points to the same tangent space and compute the covariance there. If  $\phi$  is chosen to be the exponential map, then we would use the logarithmic map for its inverse.

Another significant departure compared to Simon’s UKF is that we repeat the above steps for the noise vector  $w_k$  as well to obtain a propagated noise covariance instead of simply adding  $Q_{k+1}$  to the propagated  $P^-$ . This is to make sure that the process noise goes through the same retraction and ends up in the same tangent space.

Now that we have finished the time update, the rest of the UKF can be used nearly verbatim for UKF-M. The only change is that we replace difference of state estimates with its analog  $\xi(\hat{x}_{k+1}^-)$ .

Here is the full algorithm:

## 4 Experiments

We demonstrate the performance of UKF-M through a simulated experiment: 2D robot localization []. The goal is to estimate the direction and position (pose) of a robot moving at a constant speed on a circular trajectory. The process noise comes from the motors, which provide forward speed and angular velocity as inputs. This is an excellent task for UKF-M because any 2D direction can be described by a simple rotation matrix, which is in the Lie group  $SO(2) \cong S^1$ , one of the simplest Lie groups that gives non-trivial applications. Clearly  $S^1$  is not a vector space, but as long as we are dealing with small angles, we can treat them like vectors. This gives us the justification to use UKF as well.

The direction and position together lie on the 3-dimensional manifold  $M = SO(2) \times \mathbb{R}^2$ . Its tangent space is thus also 3-dimensional.

Now we describe the setup of the experiment.

- (1) The state is an element of  $SO(2) \times \mathbb{R}^2$ . Therefore,

$$x = \begin{pmatrix} \theta \\ a \\ b \end{pmatrix}, \text{ or equivalently, } x = \left( \text{Rot}(\theta), \begin{pmatrix} a \\ b \end{pmatrix} \right),$$

where  $\theta \in S^1$  or  $\text{Rot}(\theta) \in SO(2)$  describes the direction of the body frame and  $\begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^2$  describes the position in the 2D plane. Note that we use the first expression for UKF and the second expression for UKF-M.

- (2) The robot starts at  $\theta = 0$ ,  $(a, b) = (25, 0)$ , travels counterclockwise at a constant forward speed, and traces out a circle of radius  $r = 25$  meters as the true trajectory. Its direction is always forward and tangent to the circle.
- (3) The inputs are the forward speed and angular speed provided by the motor. In order to trace out a circle, we assume that the robot receives nominal inputs  $u_0$  of constant forward speed  $s = 2\pi r/T$  and angular speed  $\omega = 2\pi/T$ , where  $T$  is the total time. This way, the robot completes a single loop at the end of the experiment. We assume that the input errors  $w$  from the motor have small standard deviations of  $\frac{\pi}{180}$  rad/s,

0.01 m/s, and 0.01 m/s in the body frame. Then the true input in the body frame is

$$u_B = u_0 + w = \begin{pmatrix} \omega + w(1) \\ w(2) \\ s + w(3) \end{pmatrix}.$$

We can use the direction matrix  $C = \text{Rot}(\theta)$  of the robot to convert this to inertial frame, denoted  $u$ . Then we have the continuous dynamics

$$\dot{x} = \begin{pmatrix} \dot{\theta} \\ \dot{a} \\ \dot{b} \end{pmatrix} + u.$$

- (4) Since the input is digital and thus discrete, we shall discretize the dynamics. Let the frequency of input be 100 Hz. Then  $dt = 0.01$  s. We assume input to be constant over each  $dt$ , so the discretized dynamics  $f(x_k, u_0, w_k)$  is

$$x_{k+1} = (\text{Rot}[(\omega + w_k(1))dt], x_k(2:3) + \text{Rot}(\theta_k)[u_0 + w_k(2:3)]dt) \quad (5)$$

- (5) We only measure the position of the robot using GPS, with a frequency of 1 Hz and a standard deviation of 5 meters in each coordinate in the inertial frame. Thus,  $R = \begin{pmatrix} 25 & 0 \\ 0 & 25 \end{pmatrix}$  and measurement noise  $v \sim (0, R)$ .  $\Delta t = 1$  s. The measurement function  $h(x_i, v_i)$  is

$$y_i = x_i(2:3) + v_i.$$

Here we use a different index to emphasize that measurements are taken at much less frequency than inputs.

- (6) The initial estimate is

$$\hat{x}_0 = \begin{pmatrix} \frac{\pi}{180} \\ 24.9 \\ 0.1 \end{pmatrix}$$

- (7) We have high confidence in our initial estimate, so we let

$$P_0 = \text{diag}\left(\left(\frac{\pi}{180}\right)^2, 0.01, 0.01\right)$$

It remains to define the retraction map  $\phi : (SO(2) \times \mathbb{R}^2) \times \mathbb{R}^3 \rightarrow SO(2) \times \mathbb{R}^2$  and its inverse. We shall use the exponential maps of  $SO(2)$  and  $\mathbb{R}^2$ . We have already described the exponential map of  $SO(2)$ . The exponential map of  $\mathbb{R}^2$  reduces to vector addition. Thus  $\phi$  is defined as

$$\phi : \left( \left[ \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix} \right], \begin{pmatrix} \xi_\theta(\hat{x}) \\ \xi_a(\hat{x}) \\ \xi_b(\hat{x}) \end{pmatrix} \right) \mapsto \left( \text{Rot}(\xi_\theta(\hat{x})) \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \begin{pmatrix} a + \xi_a(\hat{x}) \\ b + \xi_b(\hat{x}) \end{pmatrix} \right). \quad (6)$$

The inverse of  $\phi$  is defined as

$$\phi^{-1} : \left( \left[ C, \begin{pmatrix} a \\ b \end{pmatrix} \right], \left[ D, \begin{pmatrix} c \\ d \end{pmatrix} \right] \right) \mapsto \left[ \log_C(D), \begin{pmatrix} c - a \\ d - b \end{pmatrix} \right]. \quad (7)$$

## 5 Results