Directional Statistics and Filtering Using libDirectional

Gerhard Kurz
Karlsruhe Institute of Technology (KIT)

Igor Gilitschenski
Massachusetts Institute of Technology

Florian Pfaff
Karlsruhe Institute of Technology (KIT)

Lukas Drude
University of Paderborn

Uwe D. Hanebeck
Karlsruhe Institute of Technology (KIT)

Reinhold Haeb-Umbach
University of Paderborn

Roland Y. Siegwart
ETH Zürich

Abstract

In this paper, we present **libDirectional**, a MATLAB library for directional statistics and directional estimation. It supports a variety of commonly used distributions on the unit circle, such as the von Mises, wrapped normal, and wrapped Cauchy distributions. Furthermore, various distributions on higher-dimensional manifolds such as the unit hypersphere and the hypertorus are available. Based on these distributions, several recursive filtering algorithms in **libDirectional** allow estimation on these manifolds. The functionality is implemented in a clear, well-documented, and object-oriented structure that is both easy to use and easy to extend.

Keywords: recursive filtering, phase estimation, orientation estimation, circle, hypertorus, hypersphere.

1. Introduction

Directional statistics is a subfield of statistics that deals with quantities defined on manifolds such as the unit circle or the unit hypersphere. Originally mostly developed with geoscientific applications in mind (Mardia 1981; Bingham 1974; Gaile and Burt 1980), directional statistics has gained widespread interest in various areas during the past decades, for example in biology (Batschelet 1981; Mardia, Taylor, and Subramaniam 2007), robotics (Glover and Kaelbling 2014; Feiten, Lang, and Hirche 2013; Markovic, Chaumette, and Petrovic 2014), machine

learning (Banerjee, Dhillon, Ghosh, and Sra 2005; Gopal and Yang 2014; Diethe, Twomey, and Flach 2015), aerospace (Horwood and Poore 2014; Darling and DeMars 2015; Kurz and Hanebeck 2017), and signal processing (Traa and Smaragdis 2013; Azmani, Reboul, Choquel, and Benjelloun 2009; Drude, Chinaev, Vu, and Haeb-Umbach 2014). A good introduction to the topic can, for example, be found in the book by Mardia and Jupp (1999).

There are a number of software packages that implement methods stemming from directional statistics (see Section 2). While some of these packages provide good implementations of certain algorithms, most of them are limited to few or just a single probability distribution. Also, usually only a single type of manifold is considered. Moreover, most software packages do not include the ability to perform recursive filtering. To remedy these deficiencies, we present a new software package called **libDirectional**.

libDirectional is a library written in MATLAB (The MathWorks Inc. 2017), a very popular programming language in the engineering community. A few functions are written in C or C++ for performance reasons but they can still be conveniently called from MATLAB. The design of the library follows an object-oriented approach, which makes it user-friendly and easily extensible. As the code is thoroughly documented, the library is simple to use, modify, and extend.

We designed **libDirectional** with several goals in mind. It is intended to allow the user to easily and quickly experiment with different directional probability densities and filters. Thus, we think it is a valuable tool not only for learning but also for teaching directional statistics. Furthermore, **libDirectional** allows rapid prototyping of directional algorithms for a variety of applications. Finally, one of the important goals of the library is to facilitate an easy comparison of different algorithms, for example the quantitative evaluation of a variety of filters.

While it is almost impossible to ensure that any non-trivial software is completely free of bugs, we put a strong emphasis on correctness in the development of **libDirectional**. In particular, we implemented a large number of unit tests that can be used to automatically test most of the implemented features and serve as additional usage examples. Aside from the unit tests, we include a lot of assertions (Hoare 2003) in the code to reduce the risk of problems, for example inadvertently calling certain functions with invalid parameters such as a vector of incorrect dimension or a covariance matrix lacking symmetry or positive definiteness. Even though these assertions introduce some overhead, we decided that early detection of errors and ease of use are more important than speed for **libDirectional**. In case the code is used in a real-time environment where speed is critical, it is still possible to remove certain checks to reduce this overhead.

2. Related work

Over the course of the past two decades, a number of software packages for directional statistics have been developed and published. In this section, we give an overview of the most significant packages.

A lot of the software developed for directional statistics only considers the circular case. An early example is **CIRCSTAT**, a collection of Stata (StataCorp 2017) programs for circular statistics developed by Cox (1998). Later, the well-known book on circular statistics by Jammalamadaka and Sengupta (2001) was released that includes a floppy disk containing

CircStats, a library written in S-PLUS (Insightful Corp. 2003) by Ulric Lund. The package **CircStats** was later ported to R. An enhanced version of this package was subsequently published under the name **circular** (Lund and Agostinelli 2018). This package is discussed in more detail in the book by Pewsey, Neuhäuser, and Ruxton (2013). Other R packages such as **isocir** (Barragán, Fernández, Rueda, and Peddada 2013), a package for isotonic inference for circular data, and **NPCirc** (Oliveira, Crujeiras, and Rodríguez-Casal 2014), a package implementing nonparametric circular regression, were built on top of **circular**. It should be noted that **NPCirc** also includes the ability to perform circular-circular regression (on the torus) and circular-linear regression (on the cylinder). There are also some packages for other programming languages. A MATLAB toolbox called **CircStat** (not to be confused with the aforementioned packages **CIRCSTAT** and **CircStats**) was published by Berens (2009). As MATLAB is very popular in the engineering community, we have chosen this language for **libDirectional** as well. There have also been articles with associated code for circular statistics in C++ (Krogan 2011) and Fortran (Allinger 2013). Furthermore, there is a closed-source commercial software called **Oriana** for circular statistics by Kovach Computing Services (2011).

While there is quite a lot of software available for the circular case, there are only few packages that deal with hyperspherical data. An early example is **SPAK** (Leong and Carlile 1998), a package written in MATLAB that deals with Kent distributions (Kent 1982) and offers quite limited functionality. A fairly comprehensive library for the Bingham distribution named **libBingham** was published by Glover (2013). It is written in both MATLAB and C, and can be used from either language. An R package called **movMF** that deals with mixtures of von Mises-Fisher distributions was later published by Hornik and Grün (2014).

Some software for handling orientations has also been published. The Bingham-based library **libBingham** mentioned above is capable of handling rotations using a quaternion representation. Moreover, there is an R package called **orientlib** by Murdoch (2003) and another R package called **rotations** by Stanfill, Hofmann, and Genschel (2014).

Although the software listed above is very useful for a variety of problems and has successfully been used by many scientists, there are some deficiencies that we seek to address in **libDirectional**. Most state-of-the-art software packages are limited to just a single manifold (most frequently the circle) and in some cases to just one particular probability distribution. While this may be fine for scientists only interested in a particular manifold or a particular distribution, in many applications, data on multiple different manifolds is to be considered and more than one probability distribution is of interest. Therefore, we implemented a number of common distributions defined on several manifolds in a unified manner in **libDirectional**. The second issue with most existing software is that only very few packages (e.g., **libBingham**) contain the functionality necessary for recursive estimation. As there is a significant demand for recursive filtering in many applications, e.g., in robotics, autonomous vehicles, aeronautics, etc., we provide several recursive filtering algorithms in **libDirectional**.

3. Probability distributions

In this section, we introduce the probability distributions implemented in **libDirectional**. These distributions can be classified according to the manifold on which they are defined. First, we consider distributions on the unit circle, then probability distributions on the torus, the unit hypersphere, and circular-linear spaces.

3.1. Circle

In the following, we give an overview of the distributions defined on the unit circle that are implemented in **libDirectional**. In general, we parameterize the unit circle as the half-open interval $[0, 2\pi)$ while keeping the topology of the unit circle in mind.

There are several common techniques for deriving circular probability distributions. A widely used method is called *wrapping*. We start with a real random variable $x \sim f(\cdot)$ distributed according to some probability distribution $f(\cdot)$ on $\mathbb{R}$. Now, we consider $x \bmod 2\pi$, which has the wrapped density

$$f^{\text{wrapped}}(t) = \sum_{k=-\infty}^{\infty} f(t + 2\pi k) \ . \tag{1}$$

This concept has been applied to a number of common distributions, resulting in the wrapped normal (WN), wrapped Cauchy (WC), wrapped exponential (WE), and wrapped Laplace (WL) distributions (Jammalamadaka and Kozubowski 2004; Mardia and Jupp 1999, Section 3.5.7). In some cases, the infinite sum in (1) can be simplified to a closed-form expression (e.g., for the wrapped Cauchy distribution). If a simplification is not possible, it is usually sufficient to consider a small finite number of terms of the series, see Kurz, Gilitschenski, and Hanebeck (2014c).

Another common concept consists in restricting a linear distribution on $\mathbb{R}^2$ to the unit circle. For example, restricting a two-dimensional Gaussian distribution $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C})$ with

$$\|\boldsymbol{\mu}\| = 1 \quad \text{and} \quad \mathbf{C} = \kappa \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

to the unit circle, i.e., $\|\mathbf{x}\| = 1$, yields an (unnormalized) von Mises (VM) distribution (von Mises 1918). The von Mises distribution has been further generalized by Gatto and Jammalamadaka (2007).

Of course, it is also possible to define distributions directly on the unit circle. For example, the circular uniform distribution and distributions based on Fourier series (Willsky 1974) belong to this category. Since nontrivial conditions have to be ensured for a Fourier series to be nonnegative (Fernández-Durán 2007), we also implemented the option to approximate the square root of the probability density function (PDF) as a Fourier series. By approximating the square root, the PDF values obtained by squaring are always nonnegative and therefore valid according to Pfaff, Kurz, and Hanebeck (2015, 2016b). The square root form is used by default but its complexity is hidden from the user. The class '`FourierDistribution`' only requires one additional parameter, the number of coefficients, and can be used like any other distribution in **libDirectional**. Furthermore, we provide a class named '`CircularMixture`', which allows considering mixtures of arbitrary circular distributions, for example von Mises mixtures such as used by Markovic and Petrovic (2012). To represent distributions with a non-standard density (e.g., marginal or conditional densities of certain higher-dimensional distributions), we also offer the class '`CustomCircularDistribution`'.

Aside from the continuous circular distributions, we also consider a discrete circular distribution, which can be thought of as a set of weighted samples. In line with the concept of a Dirac mixture on $\mathbb{R}^n$ (Hanebeck, Huber, and Klumpp 2009), we refer to this distribution as a wrapped Dirac mixture (WD) distribution. For n samples $\beta_1, \ldots, \beta_n \in [0, 2\pi)$ with weights

Class name	Comment
'CircularMixture'	Mixture of arbitrary circular distributions
'CircularUniformDistribution'	See Jammalamadaka and Sengupta (2001, Section 2.2.1)
'CustomCircularDistribution'	Distribution with user-specified PDF
'FourierDistribution'	See Willsky (1974), Pfaff *et al.* (2015)
'GvMDistribution'	Generalized von Mises, see Gatto and Jammalamadaka (2007)
'PWCDistribution'	Piecewise constant (a step function), see Kurz, Pfaff, and Hanebeck (2016d)
'VMDistribution'	von Mises, see von Mises (1918)
'WCDistribution'	Wrapped Cauchy, see Jammalamadaka and Sengupta (2001)
'WDDistribution'	Wrapped Dirac mixture, see Kurz, Gilitschenski, and Hanebeck (2013)
'WEDistribution'	Wrapped exponential, see Jammalamadaka and Kozubowski (2004)
'WLDistribution'	Wrapped Laplace, see Jammalamadaka and Kozubowski (2004)
'WNDistribution'	Wrapped normal, see Schmidt (1917)

Table 1: Probability distributions on the circle.

$\gamma_1, \ldots, \gamma_n > 0$, where $\sum_{j=1}^{n} \gamma_j = 1$, we write the wrapped Dirac mixture as

$$\mathcal{WD}(x; \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n) = \sum_{k=-\infty}^{\infty} \sum_{j=1}^{n} \gamma_j \delta(x + 2\pi k - \beta_j) = \sum_{j=1}^{n} \gamma_j \delta(x - \beta_j) \ ,$$

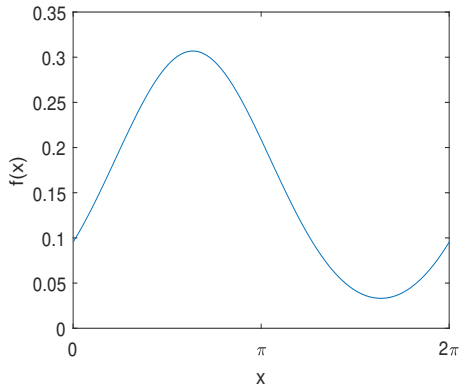
where $x \in [0, 2\pi)$ and $\delta(\cdot)$ is the Dirac delta function. Note that this distribution does not have a well-defined probability density function. It is also worth mentioning that this function does not include any wrapping terms because every Dirac component only has "probability mass" at a single point (Kurz 2015, Section 2.2.3 D).

All circular distributions implemented in **libDirectional** (see Table 1) are derived from an abstract base class called 'AbstractCircularDistribution'. This base class includes a number of functions that are applicable to all circular distributions and that are independent of the details of the particular distribution. This makes it easy to add implementations of new circular distributions, as a lot of functionality is automatically available once the PDF is defined. In particular, we offer multiple plotting functions to generate different types of visualizations.

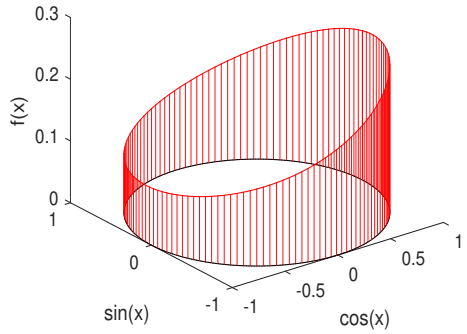
Example 1 (Plotting probability density functions) *For example, we can generate a two-dimensional plot of the PDF of a wrapped normal distribution with parameters $\mu = 2$ and $\sigma = 1.3$ simply by typing the following two commands.*

```
wn = WNDistribution(2, 1.3);
wn.plot();
```

If we also set the labels and axis using the following code, we obtain the plot depicted in Figure 1a.



(a) Two-dimensional plot of wrapped normal distribution.



(b) Three-dimensional plot of a von Mises distribution.

Figure 1: Visualizations of probability density functions on the unit circle.

```
setupAxisCircular('x');
xlabel('x'); ylabel('f(x)');
```

Similarly, we can create plots of other distributions. A three-dimensional plot of the PDF of a von Mises distribution with parameters $\mu = 6$ *and* $\kappa = 0.5$ *can be generated using the following code.*

```
vm = VMDistribution(6, 0.5);
vm.plot3d('color', 'red');
hold on; vm.plotCircle('color', 'black'); hold off;
xlabel('cos(x)'); ylabel('sin(x)'); zlabel('f(x)');
```

The resulting plot is depicted in Figure 1b. The call to `plot3d` *visualizes the density itself, whereas the call to* `plotCircle` *creates a circle in the* $\cos(x)$-$\sin(x)$*-plane.*

Also, the abstract base class contains a number of numerical methods to calculate the entropy, trigonometric moments, integrals of the PDF, etc. For most numerical methods (designated by the suffix `Numerical`), there are counterparts without this suffix that can be overridden by child classes to provide an analytical implementation. If the child class does not provide an analytical version of the algorithm, the numerical method is used as a fallback.

Example 2 (Numerical and analytical calculation) *Let us consider the wrapped normal distribution defined in Example 1 again. Suppose we want to calculate the first trigonometric moment of this distribution, i.e.,* $\mathsf{E}(\exp(ix))$*, where* $\mathsf{E}(\cdot)$ *is the expected value. For this purpose, we simply call the corresponding function:*

```
wn.trigonometricMoment(1)
```

```
ans =
  -0.1788 + 0.3906i
```

In the case of the wrapped normal distribution, `trigonometricMoment` *is a function inside the class '*`WNDistribution`*' that implements an analytic calculation of the trigonometric moment. If no analytic solution was implemented, the function* `trigonometricMoment` *in the base class '*`AbstractCircularDistribution`*' would have automatically fallen back to an algorithm based on numerical integration. Even though an analytical solution is available for the wrapped normal distribution, we can still call the numerical algorithm as follows.*

```
wn.trigonometricMomentNumerical(1)
```

```
ans =
  -0.1788 + 0.3906i
```

This can, for example, be used to compare the numerical and analytical results in order to validate the correctness of the analytical implementation. In this case, both results match up to the displayed number of digits, but in certain cases, analytical and numerical solutions may differ more significantly. Also, the numerical computation is typically slower, in some cases by several orders of magnitude.

A variety of methods are implemented for some or all circular distributions, for example the probability density function, the cumulative distribution function, the circular mean and variance, trigonometric moments, entropy, stochastic sampling, parameter estimation, conversions using trigonometric moment-matching, etc. These methods are within the code thoroughly documented, so we do not go into detail about them here.

Furthermore, we offer convolution and multiplication operations of circular probability densities for some distributions. These operations are required for the circular filtering algorithms discussed in Section 4. For example, we implement the approximations for the von Mises distribution discussed in Azmani *et al.* (2009), and the approximations for the WN distribution discussed in Kurz, Gilitschenski, and Hanebeck (2016a) and Traa and Smaragdis (2013).

One feature, however, deserves a more thorough discussion as many readers may not be familiar with it. In **libDirectional**, we have implemented several algorithms for deterministic sampling, a concept where samples are drawn from a distribution deterministically rather than stochastically. These samples are then represented using a (wrapped) Dirac mixture. The advantage of deterministic approaches is that the samples can be placed at representative positions to achieve a good representation of the true density with very few samples. This can, for example, be achieved by performing moment matching. Algorithms of this type have previously been used in Gaussian filters such as the unscented Kalman filter (UKF) by Julier and Uhlmann (2004) or the smart sampling Kalman filter (S^2KF) by Steinbring, Pander, and Hanebeck (2016). We have proposed deterministic sampling schemes based on approximation of the first trigonometric moment and the first two trigonometric moments in Kurz, Gilitschenski, and Hanebeck (2014b). Furthermore, we have proposed approximations using quantization (Gilitschenski, Kurz, Hanebeck, and Siegwart 2016a), superposition of moment-based samples, and a binary tree approximation (Kurz, Gilitschenski, Siegwart, and Hanebeck 2016c). These methods are implemented in **libDirectional** and their use is demonstrated in the following example.

Example 3 (Deterministic sampling) *Once again, we consider the wrapped normal distribution defined in Example 1. Suppose we want to approximate this distribution using a*

wrapped Dirac mixture with three components, i.e., with three samples on the unit circle. According to Kurz et al. (2014b), the approximation can preserve the first trigonometric moment. We can easily achieve this using the following command.

```
wd3 = wn.toDirac3()
```

```
wd3 =
  WDDistribution with properties:
    dim: 1
      d: [0.5740 2 3.4260]
      w: [0.3333 0.3333 0.3333]
```

The row vectors d and w correspond to the positions and weights of the Dirac components, respectively. It can be seen that the resulting wrapped Dirac mixture is evenly weighted. We can verify that the first moment is indeed preserved, similar to Example 2.

```
mwd = wd3.trigonometricMoment(1), mwn = wn.trigonometricMoment(1)
```

```
mwd =
  -0.1788 + 0.3906i
mwn =
  -0.1788 + 0.3906i
```

An approximation with five components based on the first two trigonometric moments is also possible.

```
wd5 = wn.toDirac5()
```

```
wd5 =
  WDDistribution with properties:
    dim: 1
      d: [0.1113 3.8887 1.3156 2.6844 2]
      w: [0.1855 0.1855 0.1855 0.1855 0.2581]
```

In this case, the mixture components are not evenly weighted. We can plot the resulting approximations using the following statements.

```
wn.plot(); hold on; wd3.plot('--'); wd5.plot(); hold off;
setupAxisCircular('x');
xlabel('x'); ylabel('f(x)'); legend('wn', 'wd3', 'wd5');
```

The result is depicted in Figure 2.

3.2. Torus and hypertorus

Another interesting manifold is the torus as it can be used to represent two (possibly correlated) angular quantities. We parameterize the torus as $[0, 2\pi)^2$, i.e., the Cartesian product of two circles.

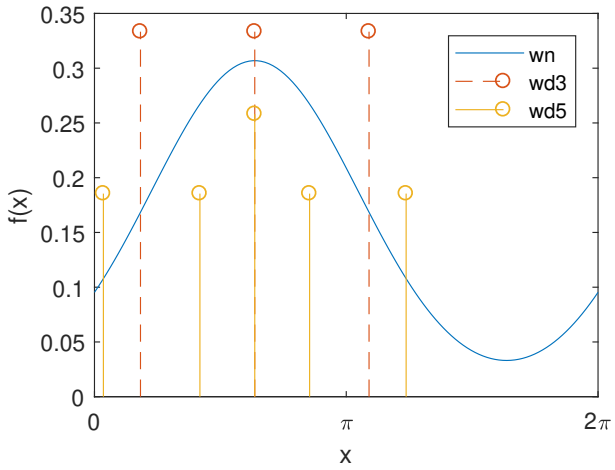


Figure 2: Example for deterministic sampling. The wrapped normal distribution is approximated with either three or five samples. Note that the height of the Dirac delta components is used to represent their weight.

On the torus, we consider several different distributions, the bivariate wrapped normal distribution, two versions of the bivariate von Mises distribution, the bivariate wrapped Dirac mixture, and a distribution based on a two-dimensional Fourier series (see Table 2). As their names suggest, these distributions constitute bivariate generalizations of the wrapped normal distribution, the von Mises distribution, the wrapped Dirac mixture, and the Fourier distribution, respectively. In the case of the wrapped normal distribution, the generalization to a higher number of dimensions is straightforward (Johnson and Wehrly 1977, Example 7.3). A bivariate wrapped normal distribution arises when a random variable distributed according to a bivariate normal distribution is wrapped in both dimensions. The bivariate wrapped Dirac distribution is obtained analogously. For the bivariate Fourier distribution, a two-dimensional Fourier series is used to approximate the density, or the square root thereof. The bivariate von Mises distribution is more tricky as there are several different, non-equivalent definitions, some of which are discussed in Mardia *et al.* (2007). In **libDirectional**, we chose to implement the sine version as well as the matrix version of the bivariate von Mises. The sine version has the advantage that it has been more thoroughly investigated than its alternatives and a number of its properties are known, e.g., a series representation for its normalization constant (Singh, Hnizdo, and Demchuk 2002). On the other hand, the matrix version has the significant advantage that it is closed under multiplication (Kurz and Hanebeck 2015b).

The overall design of toroidal distributions in **libDirectional** is similar to the circular distributions discussed above. Once again, there is an abstract base class from which all toroidal distributions inherit. Its name is '`AbstractToroidalDistribution`' and it implements a number of methods that are independent of the particular toroidal distribution. These methods can be overridden by the child classes if analytical solutions are available.

One of the key problems when dealing with toroidal distributions is the question of how to quantify correlation. Over the past decades, a number of different correlation coefficients have been proposed, and we have decided to implement several of them in **libDirectional**, namely the correlation coefficients by Johnson and Wehrly (1977), Jupp and Mardia (1980), and Jammalamadaka and Sarma (1988). More generally, for a toroidal random vector $[x_1, x_2]^\top$,

Class name	Comment
'CustomToroidalDistribution'	Allows any user-specified PDF
'ToroidalMixture'	Mixture of arbitrary toroidal distributions
'ToroidalFourierDistribution'	Bivariate Fourier distribution (Pfaff, Kurz, and Hanebeck 2016a)
'ToroidalUniformDistribution'	Uniform distribution on the torus
'ToroidalVMMatrixDistribution'	Bivariate VM, matrix version (Kurz and Hanebeck 2015b)
'ToroidalVMSineDistribution'	Bivariate VM, sine version (Singh *et al.* 2002)
'ToroidalWDDistribution'	Bivariate wrapped Dirac mixture (Kurz, Gilitschenski, Dolgov, and Hanebeck 2014a)
'ToroidalWNDistribution'	Bivariate WN (Kurz *et al.* 2014a; Kurz and Hanebeck 2015a)

Table 2: Probability distributions on the torus.

it is of interest to consider

$$\tilde{\boldsymbol{\mu}} = \mathsf{E}\left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ \cos(x_2)\\ \sin(x_2)\end{bmatrix}\right),\quad \tilde{\mathbf{C}} = \mathsf{E}\left(\left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ \cos(x_2)\\ \sin(x_2)\end{bmatrix} - \tilde{\boldsymbol{\mu}}\right)\cdot\left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ \cos(x_2)\\ \sin(x_2)\end{bmatrix} - \tilde{\boldsymbol{\mu}}\right)^{\top}\right), \tag{2}$$

which we have implemented under the name `mean4D` and `covariance4D`, respectively. These values can, for example, be used to determine the circular mean in each dimension as well as certain circular-circular correlation coefficients.

Example 4 (Bivariate wrapped normal) *First, we instantiate a bivariate wrapped normal distribution using the following statement.*

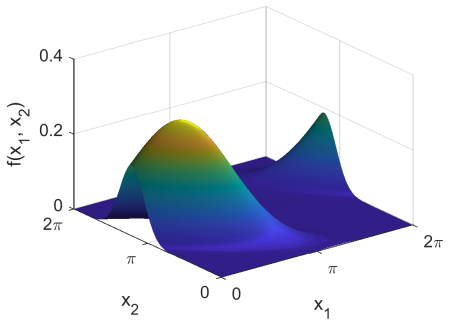
```
twn = ToroidalWNDistribution([1;3], [1, -0.8; -0.8, 0.9])
```

```
twn =
  ToroidalWNDistribution with properties:
    dim: 2
     mu: [2x1 double]
      C: [2x2 double]
```

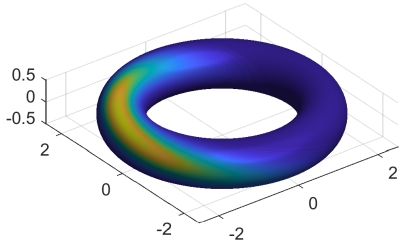
We can visualize the density of this distribution as a function $[0, 2\pi)^2 \to \mathbb{R}^+$ *using the* `plot` *method.*

```
twn.plot();
setupAxisCircular('x', 'y');
shading interp; camlight; lighting phong;
xlabel('x_1'); ylabel('x_2'); zlabel('f(x_1, x_2)');
```

The resulting plot after adjusting some MATLAB *plotting settings is shown in Figure 3a. Alternatively, we can create a visualization on the surface of a torus using the* `plotTorus` *method:*



(a) Visualization in the x_1-x_2-plane. Both x_1 and x_2 are 2π-periodic.



(b) Visualization on the torus.

Figure 3: Density of a bivariate wrapped normal distribution on the torus shown using two different visualizations.

```
twn.plotTorus();
axis equal; shading interp; camlight; lighting phong;
```

This yields the plot depicted in Figure 3b. Furthermore, we can investigate the different correlation coefficients for this distribution using the following code.

```
r1 = twn.circularCorrelationJammalamadaka(),
r2 = twn.circularCorrelationJohnson(),
r3 = twn.circularCorrelationJupp()
```

```
r1 =
   -0.8086
r2 =
   -0.8086
r3 =
   -1.0667
```

As you can see, the first correlation coefficient (Jammalamadaka and Sarma 1988) and the second correlation coefficient (Johnson and Wehrly 1977) are identical up to the displayed number of digits in this example. However, this does not hold in general. The value of the coefficient by Jupp and Mardia (1980) is quite different, and is not even restricted to the interval $[-1, 1]$.

It can be shown that for a bivariate wrapped normal distribution, the marginals are wrapped normal. They can be obtained using the following function calls.

```
wn1 = twn.marginalizeTo1D(1), wn2 = twn.marginalizeTo1D(2)
```

```
wn1 =
  WNDistribution with properties:
       mu: 1
    sigma: 1
```

Class name	Comment
'CustomHypertoroidalDistribution'	Allows any user-specified PDF
'HypertoroidalFourierDistribution'	Multivariate Fourier distribution (Pfaff *et al.* 2016a)
'HypertoroidalMixture'	Mixture of arbitrary hypertoroidal distributions
'HypertoroidalUniformDistribution'	Uniform distribution on the hypertorus
'HypertoroidalVMSineDistribution'	Multivariate VM, sine version (Mardia, Hughes, Taylor, and Singh 2008)
'HypertoroidalWDDistribution'	Multivariate wrapped Dirac mixture (Kurz 2015)
'HypertoroidalWNDistribution'	Multivariate WN (Kurz 2015)

Table 3: Probability distributions on the hypertorus.

```
      dim: 1
wn2 =
  WNDistribution with properties:
       mu: 3
    sigma: 0.9487
      dim: 1
```

As can be seen, the marginals are returned as objects of the class 'WNDistribution', i.e., a circular distribution that can be used as discussed in Section 3.1. Thus, this example illustrates one of the benefits of having implemented distributions on multiple different manifolds within a single library.

Beyond toroidal distributions, we also offer some hypertoroidal distributions, i.e., distributions on the n-torus $[0, 2\pi)^n$. An overview of the supported distributions is given in Table 3, all of which are generalizations of the corresponding toroidal distributions.

All hypertoroidal distributions use 'AbstractHypertoroidalDistribution' as their base class. Because the circle and the torus are special cases of the hypertorus for $n = 1$ and $n = 2$, respectively, circular and toroidal distributions also inherit (indirectly) from this class. In this way, a lot of code can be shared among many distributions even though they are defined on different manifolds.

3.3. Real hypersphere

In this section, we consider probability distributions defined on the unit hypersphere $S^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$, i.e., we parameterize the unit hypersphere as a set of unit vectors in $\mathbb{R}^n$ for some $n \in \mathbb{N}$. Note that this also encompasses the unit circle if $n = 2$, but uses a different parameterization (the set of two-dimensional unit vectors rather than a one-dimensional interval of length 2π) compared with the section above.

On the hypersphere, we consider several different distributions as well. The first distribution is the von Mises-Fisher distribution proposed by Fisher (1953), a generalization of the circular von Mises distribution to the unit hypersphere. It is parameterized by a unit vector $\boldsymbol{\mu}$ defining the mode of the distribution as well as a concentration parameter κ influencing its dispersion. The von Mises-Fisher distribution is unimodal and radially symmetric around $\boldsymbol{\mu}$. The Watson

Class name	Comment
'BinghamDistribution'	See Bingham (1964), Bingham (1974)
'HypersphericalDiracDistribution'	Discrete distribution on the real hypersphere
'HypersphericalUniform'	Uniform distribution on the real hypersphere
'VMFDistribution'	von Mises-Fisher distribution (Fisher 1953)
'WatsonDistribution'	See Watson (1965)
'BayesianComplexWatsonMixtureModel'	Complex Watson mixture with prior
'ComplexAngularCentralGaussian'	See Kent (1997)
'ComplexBinghamDistribution'	See Kent (1994)
'ComplexWatsonDistribution'	See Mardia and Dryden (1999)
'ComplexWatsonMixtureModel'	Mixture of complex Watson distributions

Table 4: Probability distributions on the hypersphere.

distribution (Watson 1965) is closely related and has the same set of parameters, but it is antipodally symmetric (i.e., $f(\mathbf{x}) = f(-\mathbf{x})$), and thus is bimodal with modes at $\pm\boldsymbol{\mu}$. However, it is still radially symmetric around the axis of $\boldsymbol{\mu}$. In order to represent anisotropic noise, the Watson distribution can be generalized to obtain the Bingham distribution as defined by Bingham (1974). The Bingham distribution is usually parameterized by an orthogonal matrix $\mathbf{M}$ that defines the location of the mean and the orientation of the principal axes of the uncertainty, as well as a diagonal matrix $\mathbf{Z}$ responsible for representing the uncertainties along the different axes. Furthermore, we can enforce that the diagonal entries of $\mathbf{Z}$ are sorted in ascending order and that the last diagonal entry is zero (Kurz, Gilitschenski, Julier, and Hanebeck 2014f) without changing the expressiveness of the distribution. Thus, we use this parameterization within **libDirectional**[1].

The architecture for hyperspherical probability distributions is similar to that of the previously discussed manifolds. There is a base class called '`AbstractHypersphericalDistribution`', which provides generic features independent of the particular distribution such as plotting and numerical integration. The individual distributions inherit from this class and can provide methods for the PDF, the normalization constant, stochastic sampling, parameter estimation, etc. As the normalization constant for the Bingham distribution is given by a hypergeometric function of matrix argument, it is quite expensive to evaluate. We have implemented several possible methods including the saddlepoint approximation by Kume and Wood (2005). Further discussion about the different methods for computing the normalization constant can be found in Gilitschenski, Kurz, Julier, and Hanebeck (2014b). Similar to the deterministic sampling schemes on the circle (see Example 3), we also provide a deterministic sampling scheme for the Bingham distribution, which is presented in Gilitschenski, Kurz, Julier, and Hanebeck (2016b), and a deterministic sampling scheme for the von Mises-Fisher distribution proposed in Kurz, Gilitschenski, and Hanebeck (2016b).

3.4. Complex hypersphere

This section introduces three distributions and related statistical models defined on the complex hypersphere $\mathbb{C}S^{n-1} = \{\mathbf{z} \in \mathbb{C}^n : \|\mathbf{z}\| = 1\}$. The complex Bingham distribution is defined

[1]It should be noted that some authors use slightly different parameterizations, e.g., Glover and Kaelbling (2014).

by its probability density function

$$p(\mathbf{z};\mathbf{B}) = \frac{1}{c_{\mathrm{B}}(\mathbf{B})} \exp(\mathbf{z}^{\mathsf{H}}\mathbf{B}\mathbf{z}), \quad \mathbf{z} \in \mathbb{C}S^{n-1},$$

with Hermitian transpose $\mathbf{z}^{\mathsf{H}} := \bar{\mathbf{z}}^{\top}$, where $\mathbf{z}$ is conditioned on $\mathbf{z}^{\mathsf{H}}\mathbf{z} = 1$, $c_{\mathrm{B}}(\mathbf{B})$ is an appropriate normalization term and $\mathbf{B}$ is the complex positive semi-definite parameter matrix. The complex Bingham distribution has complex symmetry, namely, it is invariant under scalar rotation, i.e., $p(\mathbf{z}) = p(\mathbf{z}\exp(i\varphi))$. Similar to the real Bingham distribution, its deviation around the mean is governed by the difference between the eigenvalues of $\mathbf{B}$. Again, the parameter matrices $\mathbf{B}$ and $\mathbf{B} + k\mathbf{I}$ define the same distribution (Kent 1994). A maximum likelihood fit according to Kent (1994) is implemented in `ComplexBinghamDistribution.fit()`.

In contrast to the real case, the complex Bingham normalization constant $c_{\mathrm{B}}(\mathbf{B})$ can be written in terms of elementary functions

$$c_{\mathrm{B}}(\mathbf{B}) = 2\pi^n \sum_{k=1}^{n} a_k \exp \lambda_k \ , \qquad a_k^{-1} = \prod_{k \neq l} (\lambda_k - \lambda_l) \ , \tag{3}$$

where λ_k are the eigenvalues of the parameter matrix $\mathbf{B}$. A symbolic implementation is used to generate code for the normalization constant `ComplexBinghamDistribution.logNorm()` and other moments of the complex Bingham distribution.

Counterintuitively, the complex Bingham distribution is a special case of the real Bingham distribution of higher dimension. The corresponding real Bingham parameter matrix can be calculated by replacing each entry $B_{kl} = \alpha_{kl}\exp(i\varphi_{kl})$ with blocks $\tilde{B}_{kl}$

$$\tilde{B}_{kl} = \alpha_{kl} \begin{pmatrix} \cos(\varphi_{kl}) & -\sin(\varphi_{kl}) \\ \sin(\varphi_{kl}) & \cos(\varphi_{kl}) \end{pmatrix}.$$

This relationship is useful to test implementations of the complex distributions against their real counterparts and is implemented in `ComplexBinghamDistribution.toReal()`. Nevertheless, the algorithms for the complex case can be implemented more efficiently without relying on the real counterpart.

The complex Watson distribution is a special case of the complex Bingham distribution. It is defined by its PDF (Mardia and Dryden 1999)

$$p(\mathbf{z};\kappa,\mathbf{w}) = \frac{1}{c_{\mathrm{W}}(\kappa)} \exp(\kappa|\mathbf{z}^{\mathsf{H}}\mathbf{w}|^2), \quad \mathbf{z} \in \mathbb{C}S^{n-1} \ ,$$

where $\mathbf{w} \in \mathbb{C}S^{n-1}$ is a complex vector with unit norm and κ governs the concentration around the mean direction. The complex Watson normalization constant $c_{\mathrm{W}}(\kappa)$ can be written in terms of elementary functions (Mardia and Dryden 1999). An implementation of the normalization constant is provided in `ComplexWatsonDistribution.logNorm()` and derivatives thereof are used for maximum likelihood estimates in `ComplexWatsonDistribution.fit()`.

Different sampling algorithms for the complex Bingham distribution are known (Kent, Constable, and Er 2004). Here, a sampling process based on sampling from truncated exponential distributions has been implemented in `ComplexBinghamDistribution.sample()`. The sampling process can be used to create samples for a complex Watson distribution and for complex Watson mixture models.

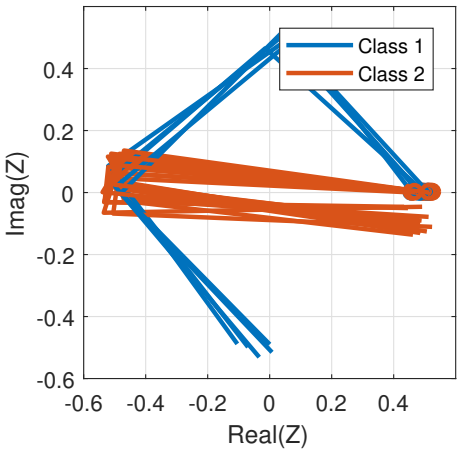


Figure 4: Samples from a complex Watson mixture model in the complex shape domain.

Early applications of the complex Watson and complex Bingham distributions are statistical modeling of two-dimensional landmarks (Kendall 1984). A fairly recent application is to describe phase and level differences in multi-channel recordings (Vu and Haeb-Umbach 2010; Drude *et al.* 2014). Different speaker positions cause different phase and level differences such that an EM algorithm for clustering can be employed. An EM algorithm for a complex Watson mixture model is implemented in `ComplexWatsonMixtureModel.fit()` (Vu and Haeb-Umbach 2010). An extension to the EM algorithm incorporates prior knowledge about the mode direction and the mixture weights of each mixture component. The corresponding variational EM algorithm is given by `BayesianComplexWatsonMixtureModel.fit()` (Drude *et al.* 2014).

Figure 4 shows some samples of a complex Watson mixture model in the complex shape domain (Kent 1994). Although PDFs on complex hyperspheres cannot be visualized easily, plots in the shape domain allow visually inspecting similarities between shapes. The unnormalized mode vectors of the underlying complex Watson distributions are

$$\mathbf{w}_1 = \begin{bmatrix} 1 & i & -1 & -i \end{bmatrix}^\top \quad \text{and} \quad \mathbf{w}_2 = \begin{bmatrix} 1+0.1i & -1+0.1i & -1-0.1i & 1-0.1i \end{bmatrix}^\top,$$

respectively.

Kent suggested the complex angular central Gaussian model as an alternative to the complex Bingham distribution. Its probability density function is given by

$$p(\mathbf{z}, \mathbf{\Sigma}) = \frac{\Gamma(D)}{2\pi^D} |\mathbf{\Sigma}|^{-1} \left(\mathbf{z}^\mathsf{H} \mathbf{\Sigma}^{-1} \mathbf{z}\right)^{-D}, \quad \mathbf{z} \in \mathbb{C}S^{n-1},$$

where $\Gamma(\cdot)$ refers to the gamma function (Kent 1997). The library provides the probability density function as well as a sampling algorithm. Additionally, parameter estimation is provided in `ComplexAngularCentralGaussian.fit()`. Natural extensions are a complex angular central Gaussian mixture model (Ito, Araki, and Nakatani 2016a) and a complex Bingham mixture model (Ito, Araki, and Nakatani 2016b), both of which found great applications in speech enhancement. We plan to add the corresponding code to **libDirectional** as future work.

3.5. SE(2)

Finally, we consider the manifold of rigid body motions in two dimensions called $SE(2)$. A rigid body motion can be seen as a rotation together with a translation. The rotation is an element of the group of two-dimensional rotations $SO(2)$ – which can be parameterized as the unit circle – and the translation is a vector in $\mathbb{R}^2$. In **libDirectional**, we consider two different continuous distributions on $SE(2)$. As they use different parameterizations of $SE(2)$, the distributions do not share a common base class.

Partially wrapped normal distribution on SE(2)

The first distribution is called the partially wrapped normal distribution (PWN) presented in Kurz, Gilitschenski, and Hanebeck (2014e) that was further discussed in Kurz (2015, Section 2.3.3)[2]. This distribution is defined on $[0, 2\pi) \times \mathbb{R}^2$ and is obtained from a normal distribution on $\mathbb{R}^3$ where the first component is wrapped. Then, the first component can be used to represent the angle of the rotation, whereas the second and third components are used to represent the translation. In analogy to the WD distribution on the circle and the bivariate WD distribution on the torus, we also define a partially wrapped Dirac mixture (PWD) distribution. Similar to the toroidal expectation values given in (2), it is of interest to consider the moments

$$\tilde{\boldsymbol{\mu}} = \mathsf{E}\left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ x_2\\ x_3\end{bmatrix}\right), \quad \tilde{\mathbf{C}} = \mathsf{E}\left(\left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ x_2\\ x_3\end{bmatrix} - \tilde{\boldsymbol{\mu}}\right) \cdot \left(\begin{bmatrix}\cos(x_1)\\ \sin(x_1)\\ x_2\\ x_3\end{bmatrix} - \tilde{\boldsymbol{\mu}}\right)^{\top}\right), \tag{4}$$

where $[x_1, x_2, x_3]^{\top}$ is a partially wrapped random variable. Note that although both consider a four-dimensional augmented random vector, (2) and (4) differ by their treatment of the last two dimensions. The values defined in (4) are available using the methods `mean4D` and `covariance4D`. Moreover, we provide the ability to obtain the marginals of a '`SE2PWNDistribution`' as a '`WNDistribution`' and as a '`GaussianDistribution`', respectively.

Modified Bingham distribution

The second distribution on $SE(2)$ is related to the Bingham distribution in the sense that it also arises by restricting a Gaussian random vector (Gilitschenski, Kurz, Julier, and Hanebeck 2014a). This is motivated by the fact that a multiplicative subgroup of dual quaternions can be used for representing elements of $SE(2)$, which is reminiscent of the approach by Matsuda, Kaji, and Ochiai (2014). Similar to the Bingham case, our distribution needs to be antipodally symmetric in order to account for the fact that unit dual quaternions are a double cover of $SE(2)$. The distribution is characterized by its PDF

$$f(\mathbf{x}) = \frac{1}{N(\mathbf{C})} \exp\left(\mathbf{x}^{\top}\, \mathbf{C}\, \mathbf{x}\right), \qquad \mathbf{x} \in S^1 \times \mathbb{R}^2,$$

where $N(\mathbf{C})$ denotes the normalization constant. This corresponds to the density of a four-dimensional Gaussian where the first two entries of $\mathbf{x}$ are interpreted as one vector that

[2] Roy, Parui, and Roy (2014) refer to this distribution as semi-wrapped Gaussian. In Lo and Willsky (1975, Equation 78), the term (n, m)-folded normal is used.

is restricted to unit length. This probability distribution is implemented within the class '`SE2BinghamDistribution`'.

Not every choice of $\mathbf{C} \in \mathbb{R}^{4\times4}$ is admissible for this distribution. In order to improve our understanding of the structure of the underlying distribution, we rewrite $\mathbf{C}$ as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2^\top \\ \mathbf{C}_2 & \mathbf{C}_3 \end{bmatrix}$$

and we can then rewrite the PDF as

$$f(\mathbf{x}_s, \mathbf{x}_t) = N(\mathbf{C})^{-1} \cdot \exp\left(\mathbf{x}_s^\top \mathbf{T}_1 \mathbf{x}_s + (\mathbf{x}_t - \mathbf{T}_2 \mathbf{x}_s)^\top \mathbf{C}_3 (\mathbf{x}_t - \mathbf{T}_2 \mathbf{x}_s)\right) ,$$

where $\mathbf{x}_s \in S^1$, $\mathbf{x}_t \in \mathbb{R}^2$ and $\mathbf{T}_1 = \mathbf{C}_1 - \mathbf{C}_2^\top \mathbf{C}_3^{-1} \mathbf{C}_2$, $\mathbf{T}_2 = -\mathbf{C}_3^{-1}\mathbf{C}_2$ with $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3 \in \mathbb{R}^{2\times2}$. From this representation, it follows that $\mathbf{C}_1$ needs to be symmetric (but not necessarily positive or negative definite), $\mathbf{C}_2$ may be arbitrary, and $\mathbf{C}_3$ has to be symmetric negative definite.

An '`SE2BinghamDistribution`' object can be constructed using either of the two constructors `SE2BinghamDistribution(C)` or `SE2BinghamDistribution(C1,C2,C3)`. Besides that, we have implemented a method for obtaining the covariance matrix using Monte Carlo integration (`computeCovarianceMCMC`), a computation of the normalization constant (`computeNC`), the mode of the density (`mode`), a parameter estimation procedure (`fit`), and deterministic as well as random sampling procedures (`sampleDeterministic` and `sample`).

4. Filters

Based on the probability distributions introduced in the previous section, it is possible to derive recursive filtering algorithms to perform recursive Bayesian estimation. In the following, we introduce a number of filters that are implemented in **libDirectional**.

4.1. Circle

Several filters for the unit circle are available in **libDirectional**. These include both filters based on circular statistics and traditional filters originally intended for linear domains that have been modified for use on the unit circle and that can be employed for comparison. In the following, we distinguish the circular filters based on the type of density they use.

WN-assumed filter

First of all, we have implemented several WN-assumed filtering algorithms in the class '`WNFilter`', i.e., filters approximating the true distribution with a WN distribution after each prediction and filtering step. This class allows predicting with a noisy identity system model, with a nonlinear measurement model in conjunction with additive noise (Kurz *et al.* 2013), and with a nonlinear measurement model in conjunction with non-additive noise. As far as the measurement model is concerned, the class can handle noisy identity measurements and nonlinear measurements given by a likelihood (Kurz, Gilitschenski, and Hanebeck 2014d) with several methods. A more thorough discussion of these scenarios can be found in Kurz *et al.* (2016a).

VM-assumed filter

Analogously, we can assume a von Mises distribution instead of a wrapped normal distribution. This alternative is implemented in '`VMFilter`'. Similar to before, we also distinguish between different types of system and measurement models as discussed in Kurz *et al.* (2016a). It should be noted that for identity system and identity measurement models, the filter proposed by Azmani *et al.* (2009) arises as a special case. Finally, we also implement a measurement update based on nonlinear measurement functions (rather than measurement likelihoods) proposed in Gilitschenski, Kurz, and Hanebeck (2015b).

Fourier filters

The Fourier identity filter and the Fourier square root filter, as explained in Pfaff *et al.* (2015, 2016a), use a truncated Fourier series to approximate the density or the square root of the density. While this filter is very universal and only makes few assumptions about the noise distributions, nonlinear measurement models require knowledge of the likelihood and for nonlinear system models, the transition density is required. Unless an identity system model with additive noise is used for the prediction step, the transition density needs to be given as one of the toroidal distributions (Pfaff *et al.* 2016b) depending on the current state and the state at the next time step. The Fourier filters are particularly powerful as we implemented the ability to approximate other distributions using the Fourier series representation described in Section 3.1. Thus, prediction and filter steps can easily be performed in an approximate fashion for arbitrary densities. For this filter, the user only needs to specify at least one additional parameter at the time of initialization. The integer n (also called `noOfCoefficients`) must be given to determine the number of Fourier coefficients. The optional string input argument `transformation` can be provided to specify if the Fourier identity filter or the Fourier square root filter should be used. If no second argument is given, the square root filter is used by default. Higher values of n result in better approximations for distributions that are more peaked but also yield a moderately higher run time as the filter has an asymptotic complexity of $O(n \log n)$ for the update step and for the prediction step with an identity model with additive noise. If a prediction step using the transition density is required, the run time complexity increases to $O(n^2 \log n)$.

Gaussian-assumed filters

In order to assess the performance of filters based on directional statistics in comparison with filters making a Gaussian assumption, we included modified versions of the unscented Kalman filter (UKF; Julier and Uhlmann 2004) based on the implementation of Steinbring (2015). For this purpose, we consider two different possibilities how the filter can be modified as discussed in Kurz *et al.* (2016a) and Kurz (2015, Section 3.1). First, we can define the filter on a (local) chart of the manifold, e.g., the open interval $(0, 2\pi)$ and try to detect and fix issues when the boundary of the chart is reached by reparameterizing if necessary. In this case, the resulting filter has a scalar state. This type of filter is implemented in the class '`CircularUKF`'. Second, it is also possible to consider a filter on the space in which the manifold is embedded and to introduce a constraint enforcing that the state always resides on the manifold. In this case, the resulting filter has a two-dimensional state vector, which is constrained to be of unit length. This type of filter is implemented in the class '`ConstrainedUKF`'.

Particle filter

A commonly used filter that avoids the Gaussian assumption is the particle filter (Arulampalam, Maskell, Gordon, and Clapp 2002). As this filter does not really depend on the underlying manifold as long as the system function and the measurement likelihood properly consider the periodicity, it is easy to adapt the particle filter to the circle. Hence, we have implemented a particle filter with sequential importance resampling (SIR) in '`CircularParticleFilter`'. The particle filter is a Markov chain Monte Carlo method, and thus, relies on random sampling, so it constitutes a nondeterministic method.

Discrete filter

We also consider a Dirac-based discrete filter based on an evenly spaced grid on the circle (Kurz *et al.* 2016d), which was previously used by Pfaff *et al.* (2015). Similar approaches have been applied to practical problems, e.g., localization of a robot (Burgard, Fox, Hennig, and Schmidt 1996). The discrete filter closely resembles the particle filter, but it uses equally spaced particles with fixed positions. Consequently, prediction and measurement update only affect the weights of the particles but not their location. The discrete filter is deterministic and can closely approximate the exact Bayesian filter provided a sufficient number of grid points is used.

Piecewise constant filter

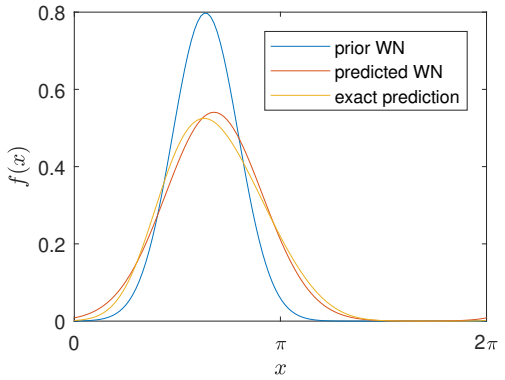
In addition to the Dirac-based discrete filter discussed above, we also offer a filter based on piecewise constant distributions (Kurz *et al.* 2016d) that is similar to a Wonham filter (Wonham 1964). This filter subdivides the interval $[0, 2\pi)$ into a predefined number of smaller intervals of equal size and assumes a uniform distribution within each interval. The filter requires a system matrix that contains the transitions probabilities from each interval into any other interval, which can be precomputed using numerical methods. For the measurement update, it is possible to use a likelihood function or to discretize the measurement space and use a precomputed measurement matrix containing the conditional probabilities of obtaining each discrete measurement given the state is in a certain interval.

Example 5 (Nonlinear circular filtering) *Let us consider a system with a circular state* $x_k \in [0, 2\pi)$ *and system dynamics*

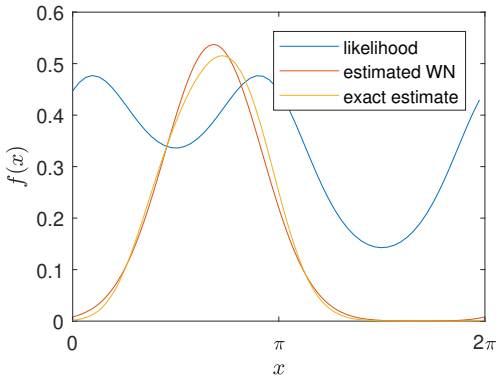
$$\begin{aligned} x_{k+1} &= a(x_k) + w_k \ , \\ a_k(x_k) &= (x_k + 0.5 \cdot \cos^2(x_k)) \mod 2\pi \ , \end{aligned}$$

where $w_k \sim \mathcal{WN}(x; 0, 0.4)$ *is WN-distributed additive noise. If we assume that the current state is distributed according to* $x_k \sim \mathcal{WN}(x; 2, 0.5)$*, we can perform the prediction step with the WN-assumed filter using the following commands.*

```
filter = WNFilter();
filter.setState(WNDistribution(2, 0.5));
a = @(x) mod(x + 0.5 * cos(x)^2, 2 * pi);
filter.predictNonlinear(a, WNDistribution(0, 0.4));
filter.getEstimate()
```



(a) Prediction step.



(b) Measurement update step.

Figure 5: Results of the nonlinear circular filtering example. We show the result obtained with the WN-assumed filter in comparison with the true result.

```
ans =
  WNDistribution with properties:
       mu: 2.1289
    sigma: 0.7377
      dim: 1
```

It can be seen that the predicted density is returned as a wrapped normal distribution (see Figure 5a). Now we consider the measurement model

$$\hat{z}_k = h_k(x_k) + v_k \ ,$$

with

$$h_k : [0, 2\pi) \to \mathbb{R}, \ h_k(x_k) = \sin(x_k) \ ,$$

where $v_k \sim \mathcal{N}(x; 0, 0.7)$ is normally distributed additive noise. In this case, we have a circular state, but a real-valued measurement. However, a circular measurement (or a measurement on a completely different manifold) would be possible as well. If we obtain a measurement, say $\hat{z} = 0.3$, we can perform the measurement update as follows.

```
h = @(x) sin(x);
measurementNoise = GaussianDistribution(0, 0.7);
likelihood = LikelihoodFactory.additiveNoiseLikelihood(h, measurementNoise);
filter.updateNonlinearProgressive(likelihood, 0.3)
filter.getEstimate()
```

```
ans =
  WNDistribution with properties:
       mu: 2.1481
    sigma: 0.7427
      dim: 1
```

Once again, we obtain the result as a wrapped normal distribution (see Figure 5b).

4.2. Torus and hypertorus

Circular filtering can be generalized to hypertoroidal filtering, just as circular distributions can be generalized to hypertoroidal distributions. As toroidal and hypertoroidal filtering is not a well-researched field, only a few filters are available. In **libDirectional**, we provide a filter called '`ToroidalWNFilter`', which is based on the toroidal wrapped normal distribution proposed in Kurz *et al.* (2014a) along with extensions for nonlinear system and measurement equations given in Kurz, Pfaff, and Hanebeck (2017). Furthermore, there is a modified version of the unscented Kalman filter (Julier and Uhlmann 2004) called '`ToroidalUKF`', which can be seen as a two-dimensional generalization of the '`CircularUKF`' discussed above. We also provide toroidal and hypertoroidal generalizations of the particle filter in '`ToroidalParticleFilter`' and '`HypertoroidalParticleFilter`', respectively. The class '`HypertoroidalFourierFilter`' offers a generalization of the Fourier filters used in the circular case. The implementation of additional hypertoroidal filters is planned as future work.

4.3. Hypersphere

In **libDirectional**, we have also included several filters for estimation on the unit hypersphere. A nonlinear filter based on the von Mises-Fisher distribution (Kurz *et al.* 2016b) is available in the class '`VMFFilter`', which inherits from the base class for hyperspherical filters called '`AbstractHypersphericalFilter`'. The class '`VMFFilter`' includes the filters by Chiuso and Picci (1998) and Markovic *et al.* (2014) as a special case. A hyperspherical particle filter and a hyperspherical version of the UKF are also available.

We have also implemented filters assuming antipodal symmetry, i.e., we assume that the points $-\mathbf{x}$ and $\mathbf{x}$ have the same probability density. This type of filter can be used for axial estimation problems, e.g., when the axis of rotation of an object is to be estimated and the direction of the axis is irrelevant. Furthermore, antipodal symmetry appears in unit quaternions, which can be used for estimation on $SO(3)$.

All hyperspherical filters with antipodal symmetry are derived from the abstract base class '`AbstractAxialFilter`'. The name refers to the estimation of an axis, e.g., a rotation axis. We provide the Bingham filter described in Kurz *et al.* (2014f), a special case of which was also considered by Glover and Kaelbling (2013). Furthermore, the Bingham filter contains the unscented extension proposed in Gilitschenski *et al.* (2016b). For comparison, **libDirectional** also includes an axial version of the Kalman filter (Kalman 1960) in the class '`AxialKalmanFilter`'. This Kalman filter uses a Gaussian distribution to approximate one of the two modes of the bimodal Bingham distribution (on one of the hemispheres).

4.4. SE(2)

We also implemented two different filters for estimation of planar rigid-body motions based on a subgroup of the unit dual quaternions that can be represented as four-dimensional vectors where the first two entries are restricted to unit length (when jointly considered as a two-dimensional vector) as proposed by Gilitschenski *et al.* (2014a). One of the filters is based on the modified Bingham distribution (in '`SE2BinghamFilter`)' and is similar to the structure of the Bingham filter (Gilitschenski, Kurz, and Hanebeck 2015a). The other filter is a UKF that is implemented (in '`SE2UKF`') in the same way as the hyperspherical UKF. The only difference is that '`SE2UKF`' ensures the resulting estimate to be a unit dual quaternion.

5. Installation and dependencies

In this section, we provide a brief explanation of the installation procedure as well as the external libraries **libDirectional** depends on.

5.1. Installation

The most recent version of **libDirectional** can always be obtained from `https://github.com/libDirectional`. In order to install the library, the entire `lib`-folder including all subdirectories has to be added to MATLAB's search path. This can be achieved using the `startup.m` script. We officially support MATLAB 2014a and later, but most functionality should be available in older versions as well. **libDirectional** is platform-independent and runs on the Windows, Linux, and Mac versions of MATLAB.

A number of functions of the library have been implemented in C++ for performance reasons. These functions can be directly called from MATLAB using the `mex`-file mechanism. In order to compile the corresponding source code, we provide the `compileAll.m` script, which compiles all files that are necessary for **libDirectional**, including the external dependencies (see Section 5.2). The compilation procedure requires a current compiler supported by MATLAB such as Microsoft Visual C++ 2013 or later[3], gcc 4.7, or Xcode.

5.2. Externals

To avoid reinventing the wheel, **libDirectional** relies on a few libraries written by other authors (see Table 5). We include all of these dependencies in the folder `externals` to make it easy for the user to run our library. Inclusion of the externals is permitted by their respective licenses and our library **libDirectional** itself is licensed under the GPL v3 license.

Eigen (Guennebaud, Jacob, and others 2010) is a C++ library for efficient matrix algorithms, which we use for some of our C++ implementations. In order to obtain high efficiency, we also use **fmath** (Shigeo 2009) to calculate exponential functions and logarithms using vector instructions from the SSE/AVX instruction set.

For the calculation of the normalization constant of the Bingham distribution, we offer several algorithms, one of which is implemented in the **mhg** library provided by Koev and Edelman (2006). Furthermore, we use the complex error function required for the multiplication of wrapped normal densities (Kurz *et al.* 2016a), which is implemented in the **Faddeeva** package by Johnson (2012). We also use a modified version of the script `circVMcdf` by Shai Revzen, which provides an implementation of the cumulative distribution function of a von Mises distribution as described in Hill (1977). Moreover, we include some code from **libBingham** by Glover (2013), but this library is not in the `external`-folder as only small parts are used and these are directly integrated into the code of **libDirectional**.

Finally, we rely on some functions of the **Nonlinear Filtering Toolbox** for MATLAB by Steinbring (2015). In particular, we use the UKF implementation as well as the deterministic sampling feature for Gaussians from this library. We provide a subset of the **Nonlinear Filtering Toolbox** within **libDirectional** that contains all required functions.

[3]There is a bug in the original version of Microsoft Visual C++ 2013 that prevents compilation. Installing Microsoft Visual Studio 2013 Update 4 resolves this issue.

Name	Author	License
`circVMcdf`	Revzen (2006)	GPL v3
Eigen	Guennebaud *et al.* (2010)	MPL 2
Faddeeva	Johnson (2012)	MIT
fmath	Shigeo (2009)	BSD (3-Clause)
libBingham	Glover (2013)	BSD (3-Clause)
mhg	Koev and Edelman (2006)	GPL v2 or later
Nonlinear Filtering Toolbox	Steinbring (2015)	GPL v3

Table 5: Dependencies of **libDirectional**.

6. Conclusion

In this paper, we have presented **libDirectional**, a MATLAB library for directional statistics and directional estimation. As we have shown, this library implements a variety of directional distributions on a number of different manifolds such as the circle, the hypertorus, and the hypersphere. Most distributions offer not only the probability density function but also algorithms for common associated problems such as visualization, parameter estimation, entropy calculation, stochastic sampling, etc. All of these methods are implemented in a clean, object-oriented design that allows the use of analytical solutions whenever possible and provides a transparent fallback to numerical solutions if analytical solutions are unavailable.

Based on these distributions, a number of different recursive filters are implemented in **libDirectional** that can be used for estimation of random variables located on the aforementioned manifolds. These filters not only include many methods based on directional statistics, but also some standard approaches that were modified for the directional setting and that can be used for comparison in order to evaluate the benefits and drawbacks of directional approaches.

We hope that the publication of **libDirectional** will make directional statistics and estimation algorithms based thereon available to a wider audience. As the library was designed to be quick to learn and easy to understand, more researches will be able to experiment with these types of methods, to apply them to various problems, and to improve upon them.

Acknowledgments

We would like to thank Jannik Steinbring for his helpful advice during the development of **libDirectional** as well as for providing prerelease versions of the **Nonlinear Estimation Toolbox** Steinbring (2015).

References

Allinger A (2013). "Circular Values Math and Statistics with Fortran." URL http://www.codeproject.com/Articles/695494/Circular-Values-Math-and-Statistics-with-FORTRAN.

Arulampalam MS, Maskell S, Gordon N, Clapp T (2002). "A Tutorial on Particle Filters

for Online Nonlinear/Non-Gaussian Bayesian Tracking." *IEEE Transactions on Signal Processing*, **50**(2), 174–188. doi:10.1109/78.978374.

Azmani M, Reboul S, Choquel JB, Benjelloun M (2009). "A Recursive Fusion Filter for Angular Data." In *IEEE International Conference on Robotics and Biomimetics (ROBIO 2009)*, pp. 882–887. doi:10.1109/robio.2009.5420492.

Banerjee A, Dhillon IS, Ghosh J, Sra S (2005). "Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions." *Journal of Machine Learning Research*, **6**, 1345–1382.

Barragán S, Fernández M, Rueda C, Peddada S (2013). "**isocir**: An R Package for Constrained Inference Using Isotonic Regression for Circular Data, with an Application to Cell Biology." *Journal of Statistical Software*, **54**(4), 1–17. doi:10.18637/jss.v054.i04.

Batschelet E (1981). *Circular Statistics in Biology*. Mathematics in Biology. Academic Press, London.

Berens P (2009). "**CircStat**: A MATLAB Toolbox for Circular Statistics." *Journal of Statistical Software*, **31**(10), 1–21. doi:10.18637/jss.v031.i10.

Bingham C (1964). *Distributions on the Sphere and on the Projective Plane*. Ph.D. thesis, Yale University.

Bingham C (1974). "An Antipodally Symmetric Distribution on the Sphere." *The Annals of Statistics*, **2**(6), 1201–1225.

Burgard W, Fox D, Hennig D, Schmidt T (1996). "Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 896–901.

Chiuso A, Picci G (1998). "Visual Tracking of Points as Estimation on the Unit Sphere." In *The Confluence of Vision and Control*, volume 237, pp. 90–105. Springer-Verlag.

Cox N (1998). "**CIRCSTAT**: Stata Modules to Calculate Circular Statistics." URL http://EconPapers.repec.org/RePEc:boc:bocode:s362501.

Darling JE, DeMars KJ (2015). "Rigid Body Attitude Uncertainty Propagation Using the Gauss-Bingham Distribution." In *25th AAS/AIAA Space Flight Mechanics Meeting*. Williamsburg.

Diethe T, Twomey N, Flach P (2015). "Bayesian Modelling of the Temporal Aspects of Smart Home Activity with Circular Statistics." In *Machine Learning and Knowledge Discovery in Databases*, pp. 279–294. Springer-Verlag, Porto.

Drude L, Chinaev A, Vu DHT, Haeb-Umbach R (2014). "Source Counting in Speech Mixtures Using a Variational EM Approach for Complex Watson Mixture Models." In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6834–6838. Florence. doi:10.1109/icassp.2014.6854924.

Feiten W, Lang M, Hirche S (2013). "Rigid Motion Estimation Using Mixtures of Projected Gaussians." In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*. Istanbul.

Fernández-Durán JJ (2007). "Models for Circular-Linear and Circular-Circular Data Constructed from Circular Distributions Based on Nonnegative Trigonometric Sums." *Biometrics*, **63**(2), 579–585. doi:10.1111/j.1541-0420.2006.00716.x.

Fisher R (1953). "Dispersion on a Sphere." *Proceedings of the Royal Society A – Mathematical, Physical and Engineering Sciences*, **217**(1130), 295–305. doi:10.1098/rspa.1953.0064.

Gaile GL, Burt JE (1980). *Directional Statistics*. Number 25 in Concepts and Techniques in Modern Geography. Geo Abstracts, University of East Anglia.

Gatto R, Jammalamadaka SR (2007). "The Generalized von Mises Distribution." *Statistical Methodology*, **4**(3), 341–353. doi:10.1016/j.stamet.2006.11.003.

Gilitschenski I, Kurz G, Hanebeck UD (2015a). "A Stochastic Filter for Planar Rigid-Body Motions." In *Proceedings of the 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2015)*. San Diego.

Gilitschenski I, Kurz G, Hanebeck UD (2015b). "Non-Identity Measurement Models for Orientation Estimation Based on Directional Statistics." In *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*. Washington, DC.

Gilitschenski I, Kurz G, Hanebeck UD, Siegwart R (2016a). "Optimal Quantization of Circular Distributions." In *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*. Heidelberg.

Gilitschenski I, Kurz G, Julier SJ, Hanebeck UD (2014a). "A New Probability Distribution for Simultaneous Representation of Uncertain Position and Orientation." In *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*. Salamanca.

Gilitschenski I, Kurz G, Julier SJ, Hanebeck UD (2014b). "Efficient Bingham Filtering Based on Saddlepoint Approximations." In *Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014)*. Beijing.

Gilitschenski I, Kurz G, Julier SJ, Hanebeck UD (2016b). "Unscented Orientation Estimation Based on the Bingham Distribution." *IEEE Transactions on Automatic Control*, **61**(1), 172–177. doi:10.1109/tac.2015.2423831.

Glover J (2013). "**libbingham** Bingham Statistics Library." http://code.google.com/p/bingham/.

Glover J, Kaelbling LP (2013). "Tracking 3-D Rotations with the Quaternion Bingham Filter." *Technical report*, MIT.

Glover J, Kaelbling LP (2014). "Tracking the Spin on a Ping Pong Ball with the Quaternion Bingham Filter." In *Proceedings of the 2014 IEEE Conference on Robotics and Automation (ICRA 2014)*. Hong Kong.

Gopal S, Yang Y (2014). "Von Mises-Fisher Clustering Models." In *Proceedings of the 31st International Conference on Machine Learning*, pp. 154–162. Beijing.

Guennebaud G, Jacob B, others (2010). "**Eigen** V3." http://eigen.tuxfamily.org.

Hanebeck UD, Huber MF, Klumpp V (2009). "Dirac Mixture Approximation of Multivariate Gaussian Densities." In *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*. Shanghai.

Hill GW (1977). "Incomplete Bessel Funcion I_0: The von Mises Distribution." *ACM Transactions on Mathematical Software*, **3**(3), 279–284. doi:10.1145/355744.355753.

Hoare CAR (2003). "Assertions: A Personal Perspective." *IEEE Annals of the History of Computing*, **25**(2), 14–25. doi:10.1109/mahc.2003.1203056.

Hornik K, Grün B (2014). "**movMF**: An R Package for Fitting Mixtures of von Mises-Fisher Distributions." *Journal of Statistical Software*, **58**(10), 1–31. doi:10.18637/jss.v058.i10.

Horwood JT, Poore AB (2014). "Gauss von Mises Distribution for Improved Uncertainty Realism in Space Situational Awareness." *SIAM/ASA Journal on Uncertainty Quantification*, **2**(1), 276–304. doi:10.1137/130917296.

Insightful Corp (2003). *S-PLUS Version 6.2*. Seattle. URL http://www.insightful.com/.

Ito N, Araki S, Nakatani T (2016a). "Complex Angular Central Gaussian Mixture Model for Directional Statistics in Mask-Based Microphone Array Signal Processing." In *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE. doi:10.1109/eusipco.2016.7760429.

Ito N, Araki S, Nakatani T (2016b). "Modeling Audio Directional Statistics Using a Complex Bingham Mixture Model for Blind Source Extraction from Diffuse Noise." In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. doi:10.1109/icassp.2016.7471718.

Jammalamadaka SR, Kozubowski TJ (2004). "New Families of Wrapped Distributions for Modeling Skew Circular Data." *Communications in Statistics – Theory and Methods*, **33**(9), 2059–2074. doi:10.1081/sta-200026570.

Jammalamadaka SR, Sarma YR (1988). "A Correlation Coefficient for Angular Variables." In K Matusita (ed.), *Statistical Theory and Data Analysis II*, pp. 349–364. North Holland, Amsterdam.

Jammalamadaka SR, Sengupta A (2001). *Topics in Circular Statistics*. World Scientific.

Johnson RA, Wehrly T (1977). "Measures and Models for Angular Correlation and Angular-Linear Correlation." *Journal of the Royal Statistical Society B*, **39**(2), 222–229. doi:10.1111/j.2517-6161.1977.tb01619.x.

Johnson SG (2012). "**Faddeeva** Package." http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package.

Julier SJ, Uhlmann JK (2004). "Unscented Filtering and Nonlinear Estimation." *Proceedings of the IEEE*, **92**(3), 401–422. doi:10.1109/jproc.2003.823141.

Jupp PE, Mardia KV (1980). "A General Correlation Coefficient for Directional Data and Related Regression Problems." *Biometrika*, **67**(1), 163–173. doi:10.2307/2335329.

Kalman RE (1960). "A New Approach to Linear Filtering and Prediction Problems." *Transactions of the ASME Journal of Basic Engineering*, **82**(1), 35–45. doi:10.1115/1.3662552.

Kendall DG (1984). "Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces." *Bulletin of the London Mathematical Society*, **16**(2), 81–121. doi:10.1112/blms/16.2.81.

Kent JT (1982). "The Fisher-Bingham Distribution on the Sphere." *Journal of the Royal Statistical Society B*, **44**(1), 71–80. doi:10.1111/j.2517-6161.1982.tb01189.x.

Kent JT (1994). "The Complex Bingham Distribution and Shape Analysis." *Journal of the Royal Statistical Society B*, **56**(2), 285–299. doi:10.1111/j.2517-6161.1994.tb01978.x.

Kent JT (1997). "Data Analysis for Shapes and Images." *Journal of Statistical Planning and Inference*, **57**(2), 181–193. doi:10.1016/s0378-3758(96)00043-2.

Kent JT, Constable PDL, Er F (2004). "Simulation for the Complex Bingham Distribution." *Statistics and Computing*, **14**(1), 53–57. doi:10.1023/b:stco.0000009414.14099.03.

Koev P, Edelman A (2006). "The Efficient Evaluation of the Hypergeometric Function of a Matrix Argument." *Mathematics of Computation*, **75**, 833–846. doi:10.1090/s0025-5718-06-01824-2.

Kovach Computing Services (2011). "**Oriana**." URL http://www.kovcomp.co.uk/oriana/index.html.

Krogan L (2011). "Circular Values Math and Statistics with C++11." URL http://www.codeproject.com/Articles/190833/Circular-Values-Math-and-Statistics-with-Cplusplus.

Kume A, Wood ATA (2005). "Saddlepoint Approximations for the Bingham and Fisher-Bingham Normalising Constants." *Biometrika*, **92**(2), 465–476.

Kurz G (2015). *Directional Estimation for Robotic Beating Heart Surgery*. Ph.D. thesis, Karlsruhe Institute of Technology, Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe, Germany.

Kurz G, Gilitschenski I, Dolgov M, Hanebeck UD (2014a). "Bivariate Angular Estimation under Consideration of Dependencies Using Directional Statistics." In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC 2014)*. Los Angeles.

Kurz G, Gilitschenski I, Hanebeck UD (2013). "Recursive Nonlinear Filtering for Angular Data Based on Circular Distributions." In *Proceedings of the 2013 American Control Conference (ACC 2013)*. Washington, DC.

Kurz G, Gilitschenski I, Hanebeck UD (2014b). "Deterministic Approximation of Circular Densities with Symmetric Dirac Mixtures Based on Two Circular Moments." In *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*. Salamanca.

Kurz G, Gilitschenski I, Hanebeck UD (2014c). "Efficient Evaluation of the Probability Density Function of a Wrapped Normal Distribution." In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*. Bonn.

Kurz G, Gilitschenski I, Hanebeck UD (2014d). "Nonlinear Measurement Update for Estimation of Angular Systems Based on Circular Distributions." In *Proceedings of the 2014 American Control Conference (ACC 2014)*. Portland.

Kurz G, Gilitschenski I, Hanebeck UD (2014e). "The Partially Wrapped Normal Distribution for SE(2) Estimation." In *Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014)*. Beijing.

Kurz G, Gilitschenski I, Hanebeck UD (2016a). "Recursive Bayesian Filtering in Circular State Spaces." *IEEE Aerospace and Electronic Systems Magazine*, **31**(3), 70–87. doi:10.1109/maes.2016.150083.

Kurz G, Gilitschenski I, Hanebeck UD (2016b). "Unscented von Mises-Fisher Filtering." *IEEE Signal Processing Letters*, **23**(4), 463–467. doi:10.1109/lsp.2016.2529854.

Kurz G, Gilitschenski I, Julier S, Hanebeck UD (2014f). "Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry." *Journal of Advances in Information Fusion*, **9**(2), 90–105.

Kurz G, Gilitschenski I, Siegwart RY, Hanebeck UD (2016c). "Methods for Deterministic Approximation of Circular Densities." *Journal of Advances in Information Fusion*, **11**(2), 138–156.

Kurz G, Hanebeck UD (2015a). "Parameter Estimation for the Bivariate Wrapped Normal Distribution." In *Proceedings of the 54th IEEE Conference on Decision and Control (CDC 2015)*. Osaka.

Kurz G, Hanebeck UD (2015b). "Toroidal Information Fusion Based on the Bivariate von Mises Distribution." In *Proceedings of the 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2015)*. San Diego.

Kurz G, Hanebeck UD (2017). "Deterministic Sampling on the Torus for Bivariate Circular Estimation." *IEEE Transactions on Aerospace and Electronic Systems*, **53**(1), 530–534. doi:10.1109/taes.2017.2650079.

Kurz G, Pfaff F, Hanebeck UD (2016d). "Discrete Recursive Bayesian Filtering on Intervals and the Unit Circle." In *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*. Baden-Baden.

Kurz G, Pfaff F, Hanebeck UD (2017). "Nonlinear Toroidal Filtering Based on Bivariate Wrapped Normal Distributions (to Appear)." In *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*. Xi'an.

Leong P, Carlile S (1998). "Methods for Spherical Data Analysis and Visualization." *Journal of Neuroscience Methods*, **80**(2), 191–200. doi:10.1016/s0165-0270(97)00201-x.

Lo J, Willsky AS (1975). "Estimation for Rotational Processes with One Degree of Freedom – Part I: Introduction and Continuous-Time Processes." *IEEE Transactions on Automatic Control*, **20**(1), 10–21. doi:10.1109/tac.1975.1100829.

Lund U, Agostinelli C (2018). ***CircStats**: Circular Statistics, from "Topics in Circular Statistics" (2001)*. R package version 0.2-6, URL https://CRAN.R-project.org/package=CircStats.

Mardia KV (1981). "Directional Statistics in Geosciences." *Communications in Statistics – Theory and Methods*, **10**(15), 1523–1543. doi:10.1080/03610928108828131.

Mardia KV, Dryden IL (1999). "The Complex Watson Distribution and Shape Analysis." *Journal of the Royal Statistical Society B*, **61**(4), 913–926. doi:10.1111/1467-9868.00210.

Mardia KV, Hughes G, Taylor CC, Singh H (2008). "A Multivariate von Mises Distribution with Applications to Bioinformatics." *Canadian Journal of Statistics*, **36**(1), 99–109. doi:10.1002/cjs.5550360110.

Mardia KV, Jupp PE (1999). *Directional Statistics*. 1st edition. John Wiley & Sons, Baffins Lane, Chichester, West Sussex, England. doi:10.1002/9780470316979.

Mardia KV, Taylor CC, Subramaniam GK (2007). "Protein Bioinformatics and Mixtures of Bivariate von Mises Distributions for Angular Data." *Biometrics*, **63**(2), 505–512. doi:10.1111/j.1541-0420.2006.00682.x.

Markovic I, Chaumette F, Petrovic I (2014). "Moving Object Detection, Tracking and Following Using an Omnidirectional Camera on a Mobile Robot." In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*. Hong-Kong.

Markovic I, Petrovic I (2012). "Bearing-Only Tracking with a Mixture of von Mises Distributions." In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pp. 707–712. doi:10.1109/iros.2012.6385600.

Matsuda G, Kaji S, Ochiai H (2014). "Anti-Commutative Dual Complex Numbers and 2D Rigid Transformation." In *Mathematical Progress in Expressive Image Synthesis I*, pp. 131–138. Springer-Verlag.

Murdoch D (2003). "**Orientlib**: An R Package for Orientation Data." *Journal of Statistical Software*, **8**(19), 1–11. doi:10.18637/jss.v008.i19.

Oliveira M, Crujeiras RM, Rodríguez-Casal A (2014). "**NPCirc**: An R Package for Nonparametric Circular Methods." *Journal of Statistical Software*, **61**(9). doi:10.18637/jss.v061.i09.

Pewsey A, Neuhäuser M, Ruxton GD (2013). *Circular Statistics in R*. Oxford University Press, Oxford.

Pfaff F, Kurz G, Hanebeck UD (2015). "Multimodal Circular Filtering Using Fourier Series." In *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*. Washington, DC.

Pfaff F, Kurz G, Hanebeck UD (2016a). "Multivariate Angular Filtering Using Fourier Series." *Journal of Advances in Information Fusion*, **11**(2), 206–226.

Pfaff F, Kurz G, Hanebeck UD (2016b). "Nonlinear Prediction for Circular Filtering Using Fourier Series." In *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*. Heidelberg.

Revzen S (2006). `circVMcdf`*: Cumulative von-Mises Distribution*. MATLAB code.

Roy A, Parui SK, Roy U (2014). "SWGMM: A Semi-Wrapped Gaussian Mixture Model for Clustering of Circular-Linear Data." *Pattern Analysis and Applications*, **18**(3), 631–645. doi:10.1007/s10044-014-0418-2.

Schmidt W (1917). "Statistische Methoden beim Gefügestudium krystalliner Schiefer." *Sitzungsberichte Akademie der Wissenschaften in Wien*, **126**, 515–539.

Shigeo M (2009). "Fast Approximate Float Function **fmath**." URL https://github.com/herumi/fmath.

Singh H, Hnizdo V, Demchuk E (2002). "Probabilistic Model for Two Dependent Circular Variables." *Biometrika*, **89**(3), 719–723.

Stanfill B, Hofmann H, Genschel U (2014). "**rotations**: An R Package for SO(3) Data." *The R Journal*, **6**(1), 68–78. doi:10.32614/rj-2014-007.

StataCorp (2017). *STATA Statistical Software: Release 15*. StataCorp LLC, College Station. URL http://www.stata.com/.

Steinbring J (2015). "**Nonlinear Estimation Toolbox**." URL https://bitbucket.org/nonlinearestimation/toolbox.

Steinbring J, Pander M, Hanebeck UD (2016). "The Smart Sampling Kalman Filter with Symmetric Samples." *Journal of Advances in Information Fusion*, **11**(1), 71–90.

The MathWorks Inc (2017). *MATLAB – The Language of Technical Computing, Version R2017b*. Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Traa J, Smaragdis P (2013). "A Wrapped Kalman Filter for Azimuthal Speaker Tracking." *IEEE Signal Processing Letters*, **20**(12), 1257–1260. doi:10.1109/lsp.2013.2287125.

von Mises R (1918). "Über die 'Ganzzahligkeit' der Atomgewichte und verwandte Fragen." *Physikalische Zeitschrift*, **XIX**, 490–500.

Vu DHT, Haeb-Umbach R (2010). "Blind Speech Separation Employing Directional Statistics in an Expectation Maximization Framework." In *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 241–244. doi:10.1109/icassp.2010.5495994.

Watson GS (1965). "Equatorial Distributions on a Sphere." *Biometrika*, **52**(1–2), 193–201.

Willsky AS (1974). "Fourier Series and Estimation on the Circle with Applications to Synchronous Communication – Part I: Analysis." *IEEE Transactions on Information Theory*, **20**(5), 577–583. doi:10.1109/tit.1974.1055280.

Wonham WM (1964). "Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering." *Journal of the Society for Industrial and Applied Mathematics A: Control*, **2**(3), 347–369. doi:10.1137/0302028.

Affiliation:

Gerhard Kurz, Florian Pfaff, Uwe D. Hanebeck
Chair for Intelligent Sensor-Actuator-Systems (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT)
Adenauerring 2
76131 Karlsruhe, Germany
E-mail: kurz.gerhard@gmail.com, florian.pfaff@kit.edu, uwe.hanebeck@ieee.org

Igor Gilitschenski
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
32 Vassar Street
Cambridge, MA 02139, United States of America
E-mail: igilitschenski@mit.edu

Lukas Drude, Reinhold Haeb-Umbach
Department of Communications Engineering
University of Paderborn
Warburger Str. 100
33098 Paderborn, Germany
E-mail: drude@nt.uni-paderborn.de, haeb@nt.uni-paderborn.de

Roland Y. Siegwart
Autonomous Systems Lab
ETH Zürich
Leonhardstrasse 21
8092 Zürich, Switzerland
E-mail: rsiegwart@ethz.ch