
clickpoints Documentation

Release 1.2.3

Richard Gerum, Sebastian Richter

Jan 24, 2020

CONTENTS

1	Installation	3
2	PenguTrack API	5
2.1	Detectors	5
2.2	Filters	11
2.3	Models	22
2.4	Stitchers	27
	Python Module Index	29
	Index	31

Click Points is a program written in the Python programming language, which serves on the one hand as an image viewer and on the other hand as an data display and annotation tool. Every frame can be annotated by a description, marked points/tracks, or marked areas (paint brush). This helps to view image data, do manual evaluation of data, help to create semi-automatic evaluation or display the results of automatic image evaluation.

INSTALLATION

Execute

```
pip install -e .
```


PENGUTRACK API

The different modules of PenguTrack.

2.1 Detectors

Module containing detector classes to be used with pengu-track filters and models.

class PenguTrack.Detectors.**AlexSegmentation** (**args, **kwargs*)
Segmentation method comparing input images to image-background buffer. Able to learn new background information.

class PenguTrack.Detectors.**AreaBlobDetector** (*object_size=1, object_number=1, threshold=None*)
Detector classifying objects by area and number to be used with pengu-track modules.

detect (*image, return_regions=False*)
Detection function. Parts the image into blob-regions with size according to their area. Returns information about the regions.

Parameters

- **image** (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.
- **return_regions** (*bool, optional*) – If True, function will return `skimage.measure.regionprops` object, else a list of the blob centroids and areas.

Returns regions (*array_like*) – List of information about each blob of adequate size.

class PenguTrack.Detectors.**AreaDetector** (*object_area=1, object_number=1, threshold=None, lower_limit=None, upper_limit=None*)
Detector classifying objects by area and number to be used with pengu-track modules.

detect (*image*)
Detection function. Parts the image into blob-regions with size according to their area. Returns information about the regions.

Parameters

- **image** (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.
- **return_regions** (*bool, optional*) – If True, function will return `skimage.measure.regionprops` object, else a list of the blob centroids and areas.

Returns regions (*array_like*) – List of information about each blob of adequate size.

class `PenguTrack.Detectors.BlobDetector` (*object_size=1, object_number=1, threshold=None*)

Detector classifying objects by size and number to be used with pengu-track modules.

detect (*image*)

Detection function. Parts the image into blob-regions with size according to *object_size*. Returns information about the regions.

Parameters *image* (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.

Returns *regions* (*array_like*) – List of information about each blob of adequate size.

class `PenguTrack.Detectors.BlobSegmentation` (*max_size, min_size=1, init_image=None*)

Segmentation method detecting blobs.

detect (*image, do_neighbours=True, *args, **kwargs*)

Segmentation function. This compares the input image to the background model and returns a segmentation map.

Parameters

- *image* (*array_like*) – Input Image.
- *do_neighbours* (*bool, optional*) – If True neighbouring pixels will be updated accordingly to their foreground vicinity, else this time-intensiv calculation will not be done.

Returns *SegMap* (*array_like, bool*) – The segmented image.

class `PenguTrack.Detectors.Detector`

This Class describes the abstract function of a detector in the pengu-track package. It is only meant for subclassing.

class `PenguTrack.Detectors.DumbViBeSegmentation` (**args, **kwargs*)

class `PenguTrack.Detectors.EmperorDetector` (*initial_image, **kwargs*)

class `PenguTrack.Detectors.FlowDetector` (*flow=None, pyr_scale=0.5, levels=3, winsize=15, iterations=3, poly_n=5, poly_sigma=1.2, flags=0, *args, **kwargs*)

class `PenguTrack.Detectors.MeanViBeSegmentation` (*sensitivity=1, n=20, m=1, init_image=None, *args, **kwargs*)

class `PenguTrack.Detectors.Measurement` (*log_probability, position, cov=None, data=None, frame=None, track_id=None*)

Base Class for detection results.

class `PenguTrack.Detectors.MoGSegmentation` (*n=10, init_image=None*)

Segmentation method assuming that pixel states depend on various gaussian distributions.

detect (*image, *args, **kwargs*)

Segmentation function. This function binarizes the input image by assuming the pixels, which do not fit the background gaussians are foreground.

Parameters *image* (*array_like*) – Image to be segmented.

Returns *SegMap* (*array_like, bool*) – Segmented Image.

segmentate (*image, *args, **kwargs*)

Segmentation function. This function binarizes the input image by assuming the pixels, which do not fit the background gaussians are foreground.

Parameters *image* (*array_like*) – Image to be segmented.

Returns `SegMap (array_like, bool)` – Segmented Image.

class `PenguTrack.Detectors.MoGSegmentation2` (*n=20, r=15, init_image=None*)

Segmentation method comparing input images to image-background buffer. Able to learn new background information.

detect (*image, do_neighbours=True, *args, **kwargs*)

Segmentation function. This compares the input image to the background model and returns a segmentation map.

Parameters

- **image** (*array_like*) – Input Image.
- **do_neighbours** (*bool, optional*) – If True neighbouring pixels will be updated accordingly to their foreground vicinity, else this time-intensiv calculation will not be done.

Returns `SegMap (array_like, bool)` – The segmented image.

class `PenguTrack.Detectors.NKCellDetector`

class `PenguTrack.Detectors.NKCellDetector2`

class `PenguTrack.Detectors.RegionPropDetector` (*RegionFilters*)

class `PenguTrack.Detectors.Segmentation`

This Class describes the abstract function of a image-segmentation-algorithm in the pengu-track package. It is only meant for subclassing.

class `PenguTrack.Detectors.SimpleAreaDetector` (*object_area=1, object_number=1, threshold=None, lower_limit=None, upper_limit=None*)

Detector classifying objects by area and number to be used with pengu-track modules.

detect (*image*)

Detection function. Parts the image into blob-regions with size according to their area. Returns information about the regions.

Parameters **image** (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.

Returns **regions** (*array_like*) – List of information about each blob of adequate size.

class `PenguTrack.Detectors.SimpleAreaDetector2` (*object_area=1, object_number=1, threshold=None, lower_limit=None, upper_limit=None, distxy_boundary=10, distz_boundary=21*)

Detector classifying objects by area and number to be used with pengu-track modules.

detect (*image, mask*)

Detection function. Parts the image into blob-regions with size according to their area. Returns information about the regions.

Parameters **image** (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.

Returns **regions** (*array_like*) – List of information about each blob of adequate size.

class `PenguTrack.Detectors.TCellDetector`

Detector classifying objects by area and number to be used with pengu-track modules.

class `PenguTrack.Detectors.ThresholdSegmentation` (*treshold, reskale=True*)

```
class PenguinTrack.Detectors.TinaCellDetector (disk_size=0, minimal_area=57, maxi-  
mal_area=350, threshold=0.2)
```

```
detect (minProj, minIndices, maxProj, maxIndices)
```

Parameters

- **minProj** –
- **minIndices** –
- **maxProj** –
- **maxIndices** –

Returns

```
class PenguinTrack.Detectors.VarianceSegmentation (treshold, r)
```

```
class PenguinTrack.Detectors.ViBeSegmentation (n=20, r=15, n_min=1, phi=16,  
init_image=None)
```

Segmentation method comparing input images to image-background buffer. Able to learn new background information.

```
detect (image, do_neighbours=True, *args, **kwargs)
```

Segmentation function. This compares the input image to the background model and returns a segmentation map.

Parameters

- **image** (*array_like*) – Input Image.
- **do_neighbours** (*bool, optional*) – If True neighbouring pixels will be updated accordingly to their foreground vicinity, else this time-intensiv calculation will not be done.

Returns **SegMap** (*array_like, bool*) – The segmented image.

```
class PenguinTrack.Detectors.WatershedDetector (object_size=1, object_number=1, thresh-  
old=None)
```

Detector classifying objects by area and number. It uses watershed algorithms to depart bigger areas. To be used with pengu-track modules.

```
detect (image, return_regions=False)
```

Detection function. Parts the image into blob-regions with size according to their area. Then departs bigger areas into smaller ones with watershed method. Returns information about the regions.

Parameters

- **image** (*array_like*) – Image will be converted to uint8 Greyscale and then binnearized.
- **return_regions** (*bool, optional*) – If True, function will return `skimage.measure.regionprops` object, else a list of the blob centroids and areas.

Returns *list* – List of information about each blob of adequate size.

```
class PenguinTrack.Detectors.dotdict  
enables dot access on dicts
```

```
PenguinTrack.Detectors.extended_regionprops (label_image, intensity_image=None,  
cache=True)
```

Measure properties of labeled image regions.

Parameters

- **label_image** $((N, M) \text{ ndarray})$ – Labeled input image. Labels with value 0 are ignored.
- **intensity_image** $((N, M) \text{ ndarray}, \text{optional})$ – Intensity (i.e., input) image with same size as labeled image. Default is None.
- **cache** $(\text{bool}, \text{optional})$ – Determine whether to cache calculated properties. The computation is much faster for cached properties, whereas the memory consumption increases.

Returns **properties** (*list of RegionProperties*) – Each item describes one labeled region, and can be accessed using the attributes listed below.

Notes

The following properties can be accessed as attributes or keys:

area [int] Number of pixels of region.

bbox [tuple] Bounding box $(\text{min_row}, \text{min_col}, \text{max_row}, \text{max_col})$. Pixels belonging to the bounding box are in the half-open interval $[\text{min_row}; \text{max_row})$ and $[\text{min_col}; \text{max_col})$.

bbox_area [int] Number of pixels of bounding box.

centroid [array] Centroid coordinate tuple (row, col) .

convex_area [int] Number of pixels of convex hull image.

convex_image $[(H, J) \text{ ndarray}]$ Binary convex hull image which has the same size as bounding box.

coords $[(N, 2) \text{ ndarray}]$ Coordinate list (row, col) of the region.

eccentricity [float] Eccentricity of the ellipse that has the same second-moments as the region. The eccentricity is the ratio of the focal distance (distance between focal points) over the major axis length. The value is in the interval $[0, 1)$. When it is 0, the ellipse becomes a circle.

equivalent_diameter [float] The diameter of a circle with the same area as the region.

euler_number [int] Euler characteristic of region. Computed as number of objects (= 1) subtracted by number of holes (8-connectivity).

extent [float] Ratio of pixels in the region to pixels in the total bounding box. Computed as $\text{area} / (\text{rows} * \text{cols})$

filled_area [int] Number of pixels of filled region.

filled_image $[(H, J) \text{ ndarray}]$ Binary region image with filled holes which has the same size as bounding box.

image $[(H, J) \text{ ndarray}]$ Sliced binary region image which has the same size as bounding box.

inertia_tensor $[(2, 2) \text{ ndarray}]$ Inertia tensor of the region for the rotation around its mass.

inertia_tensor_eigvals [tuple] The two eigen values of the inertia tensor in decreasing order.

intensity_image [ndarray] Image inside region bounding box.

label [int] The label in the labeled input image.

local_centroid [array] Centroid coordinate tuple (row, col) , relative to region bounding box.

major_axis_length [float] The length of the major axis of the ellipse that has the same normalized second central moments as the region.

max_intensity [float] Value with the greatest intensity in the region.

mean_intensity [float] Value with the mean intensity in the region.

min_intensity [float] Value with the least intensity in the region.

minor_axis_length [float] The length of the minor axis of the ellipse that has the same normalized second central moments as the region.

moments [(3, 3) ndarray] Spatial moments up to 3rd order:

```
m_ji = sum{ array(x, y) * x^j * y^i }
```

where the sum is over the x, y coordinates of the region.

moments_central [(3, 3) ndarray] Central moments (translation invariant) up to 3rd order:

```
mu_ji = sum{ array(x, y) * (x - x_c)^j * (y - y_c)^i }
```

where the sum is over the x, y coordinates of the region, and x_c and y_c are the coordinates of the region's centroid.

moments_hu [tuple] Hu moments (translation, scale and rotation invariant).

moments_normalized [(3, 3) ndarray] Normalized moments (translation and scale invariant) up to 3rd order:

```
nu_ji = mu_ji / m_00^[(i+j)/2 + 1]
```

where m_{00} is the zeroth spatial moment.

orientation [float] Angle between the X-axis and the major axis of the ellipse that has the same second-moments as the region. Ranging from $-\pi/2$ to $\pi/2$ in counter-clockwise direction.

perimeter [float] Perimeter of object which approximates the contour as a line through the centers of border pixels using a 4-connectivity.

solidity [float] Ratio of pixels in the region to pixels of the convex hull image.

weighted_centroid [array] Centroid coordinate tuple (`row`, `col`) weighted with intensity image.

weighted_local_centroid [array] Centroid coordinate tuple (`row`, `col`), relative to region bounding box, weighted with intensity image.

weighted_moments [(3, 3) ndarray] Spatial moments of intensity image up to 3rd order:

```
wm_ji = sum{ array(x, y) * x^j * y^i }
```

where the sum is over the x, y coordinates of the region.

weighted_moments_central [(3, 3) ndarray] Central moments (translation invariant) of intensity image up to 3rd order:

```
wmu_ji = sum{ array(x, y) * (x - x_c)^j * (y - y_c)^i }
```

where the sum is over the x, y coordinates of the region, and x_c and y_c are the coordinates of the region's weighted centroid.

weighted_moments_hu [tuple] Hu moments (translation, scale and rotation invariant) of intensity image.

weighted_moments_normalized [(3, 3) ndarray] Normalized moments (translation and scale invariant) of intensity image up to 3rd order:

```
wnu_ji = wmu_ji / wm_00^[(i+j)/2 + 1]
```

where wm_{00} is the zeroth spatial moment (intensity-weighted area).

Each region also supports iteration, so that you can do:

```
for prop in region:
    print(prop, region[prop])
```

2.2 Filters

Module containing filter classes to be used with pengu-track detectors and models.

class `PenguTrack.Filters.AdaptedKalmanFilter` (*model*, *evolution_variance*, *measurement_variance*, ***kwargs*)

predict (*u=None*, *i=None*)

Function to get predictions from the corresponding model. Handles time-stamps and control-vectors.

Parameters

- **u** (*array_like*, *optional*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

update (*z=None*, *i=None*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguTrack.Measurement*, *optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

class `PenguTrack.Filters.AdvancedKalmanFilter` (**args*, ***kwargs*)

This Class describes a advanced, self tuning version of the kalman-filter in the pengu-track package. It calculates actual believed values from predictions and measurements.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **Lag** (*int*) – The number of state-vectors, which are taken into account for the self-tuning algorithm.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.

- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.
- **A** (*np.asarray*) – State-Transition-Matrix, describing the evolution of the state without fluctuation. Received from the physical Model.
- **B** (*np.asarray*) – State-Control-Matrix, describing the influence of external-control on the states. Received from the physical Model.
- **C** (*np.asarray*) – Measurement-Matrix, describing the projection of the measurement-vector from the state-vector. Received from the physical Model.
- **G** (*np.asarray*) – Evolution-Matrix, describing the evolution of state vectors by fluctuations from the state-distribution. Received from the physical Model.
- **Q** (*np.asarray*) – Covariance matrix (time-evolving) for the state-distribution.
- **Q_0** (*np.asarray*) – Covariance matrix (initial state) for the state-distribution.
- **R** (*np.asarray*) – Covariance matrix (time-evolving) for the measurement-distribution.
- **R_0** (*np.asarray*) – Covariance matrix (initial state) for the measurement-distribution.
- **P_0** (*np.asarray*) – Covariance matrix (initial state) for the believe-distribution.

update (*args, **kwargs)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguTrack.Measurement*, *optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

```
class PenguTrack.Filters.Filter (model,          no_dist=False,      meas_dist=<MagicMock
                                name='mock.stats.uniform()'      id='139792175666512'>,
                                state_dist=<MagicMock             name='mock.stats.uniform()'
                                id='139792175666512'>,      prob_update=True,      *args,
                                **kwargs)
```

This Class describes the abstract function of a filter in the pengu-track package. It is only meant for subclassing.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.

- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.

downdate (*t=None*)

Function erases the time-step *t* from the Measurements and Believes.

Parameters *t* (*int*, *optional*) – Time-steps for which filtering should be erased.

downfilter (*t=None*)

Function erases the timestep *t* from the class dictionaries (Measurement, X, Prediction)

Parameters *t* (*int*, *optional*) – Time-steps for which filtering should be erased.

filter (*u=None*, *z=None*, *i=None*)

Function to get predictions from the model and update the same timestep *i*.

Parameters

- *u* (*array_like*, *optional*) – Recent control-vector.
- *z* (*PenguTrack.Measurement*, *optional*) – Recent measurement.
- *i* (*int*) – Recent/corresponding time-stamp

Returns

- *u* (*array_like*, *optional*) – Recent control-vector.
- *z* (*PenguTrack.Measurement*) – Recent measurement.
- *i* (*int*) – Recent/corresponding time-stamp.

fit (*u*, *z*)

Function to auto-evaluate all measurements *z* with control-vectors *u* and starting probability *p*. It returns the believed values *x*, the corresponding probabilities *p* and the predictions *x_tilde*.

Parameters

- *u* (*array_like*) – List of control-vectors.
- *z* (*array_like*) – List of measurement-vectors.

predict (*u=None*, *i=None*)

Function to get predictions from the corresponding model. Handles time-stamps and control-vectors.

Parameters

- *u* (*array_like*, *optional*) – Recent control-vector.
- *i* (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

unpredict (*t=None*)

Function erases the time-step *t* from the Believes, Predictions and Controls.

Parameters **t** (*int, optional*) – Time-steps for which filtering should be erased.

update (*z=None, i=None*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguTrack.Measurement, optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

class PenguTrack.Filters.**HungarianTracker** (*_filter, model, *args, **kwargs*)

class PenguTrack.Filters.**HybridSolver** (**args, **kwargs*)

This Class describes a filter, which is capable of assigning measurements to tracks, which again are represented by sub-filters. The type of these can be specified, as well as a physical model for predictions. With these objects it is possible to assign possibilities to combinations of measurement and prediction.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **Filter_Class** (*PenguTrack.Filter object*) – A Type of Filter from which all subfilters should be built.
- **Filters** (*dict*) – Dictionary containing all sub-filters as PenguTrack.Filter objects.
- **Active_Filters** (*dict*) – Dictionary containing all sub-filters, which are currently updated.
- **FilterThreshold** (*int*) – Number of time steps, before a filter is set inactive.
- **LogProbabilityThreshold** (*float*) – Threshold, under which log-probabilities are concerned negligible.
- **filter_args** (*list*) – Filter-Type specific arguments from the Multi-Filter initialisation can be stored here.
- **filter_kwargs** (*dict*) – Filter-Type specific keyword-arguments from the Multi-Filter initialisation can be stored here.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.

- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.

class `PenguTrack.Filters.InformationFilter` (*model*, *evolution_variance*, *measurement_variance*, ***kwargs*)

This Class describes a kalman-filter in the pengu-track package. It calculates actual believed values from predictions and measurements.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.
- **A** (*np.asarray*) – State-Transition-Matrix, describing the evolution of the state without fluctuation. Received from the physical Model.
- **B** (*np.asarray*) – State-Control-Matrix, describing the influence of external-control on the states. Received from the physical Model.
- **C** (*np.asarray*) – Measurement-Matrix, describing the projection of the measurement-vector from the state-vector. Received from the physical Model.
- **G** (*np.asarray*) – Evolution-Matrix, describing the evolution of state vectors by fluctuations from the state-distribution. Received from the physical Model.
- **Q** (*np.asarray*) – Covariance matrix (time-evolving) for the state-distribution.
- **Q_0** (*np.asarray*) – Covariance matrix (initial state) for the state-distribution.
- **R** (*np.asarray*) – Covariance matrix (time-evolving) for the measurement-distribution.
- **R_0** (*np.asarray*) – Covariance matrix (initial state) for the measurement-distribution.

- **P_0** (*np.asarray*) – Covariance matrix (initial state) for the believe-distribution.

predict (*u=None, i=None*)

Function to get predictions from the corresponding model. Handles time-stamps and control-vectors.

Parameters

- **u** (*array_like, optional*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

update (*z=None, i=None*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguTrack.Measurement, optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

class PenguTrack.Filters.**KalmanBaseFilter** (*model, evolution_variance, measurement_variance, **kwargs*)

This Class describes a kalman-filter in the pengu-track package. It calculates actual believed values from predictions and measurements.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.

- **A** (*np.asarray*) – State-Transition-Matrix, describing the evolution of the state without fluctuation. Received from the physical Model.
- **B** (*np.asarray*) – State-Control-Matrix, describing the influence of external-control on the states. Received from the physical Model.
- **C** (*np.asarray*) – Measurement-Matrix, describing the projection of the measurement-vector from the state-vector. Received from the physical Model.
- **G** (*np.asarray*) – Evolution-Matrix, describing the evolution of state vectors by fluctuations from the state-distribution. Received from the physical Model.
- **Q** (*np.asarray*) – Covariance matrix (time-evolving) for the state-distribution.
- **Q_0** (*np.asarray*) – Covariance matrix (initial state) for the state-distribution.
- **R** (*np.asarray*) – Covariance matrix (time-evolving) for the measurement-distribution.
- **R_0** (*np.asarray*) – Covariance matrix (initial state) for the measurement-distribution.
- **P_0** (*np.asarray*) – Covariance matrix (initial state) for the believe-distribution.

```
class PenguinTrack.Filters.KalmanFilter(model, evolution_variance, measurement_variance,
                                     **kwargs)
```

predict (*u=None, i=None*)

Function to get predictions from the corresponding model. Handles time-stamps and control-vectors.

Parameters

- **u** (*array_like, optional*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

update (*z=None, i=None*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguinTrack.Measurement, optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguinTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

```
class PenguinTrack.Filters.MultiFilter(_filter, model, *args, **kwargs)
```

This Class describes a filter, which is capable of assigning measurements to tracks, which again are represented by sub-filters. The type of these can be specified, as well as a physical model for predictions. With these objects it is possible to assign possibilities to combinations of measurement and prediction.

Variables

- **Model** (*PenguinTrack.model object*) – A physical model to gain predictions from data.
- **Filter_Class** (*PenguinTrack.Filter object*) – A Type of Filter from which all subfilters should be built.

- **Filters** (*dict*) – Dictionary containing all sub-filters as `PenguTrack.Filter` objects.
- **Active_Filters** (*dict*) – Dictionary containing all sub-filters, which are currently updated.
- **FilterThreshold** (*int*) – Number of time steps, before a filter is set inactive.
- **LogProbabilityThreshold** (*float*) – Threshold, under which log-probabilities are concerned negligible.
- **filter_args** (*list*) – Filter-Type specific arguments from the Multi-Filter initialisation can be stored here.
- **filter_kwargs** (*dict*) – Filter-Type specific keyword-arguments from the Multi-Filter initialisation can be stored here.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.

downdate (*t=None*)

Function erases the time-step *t* from the Measurements and Believes.

Parameters *t* (*int, optional*) – Time-steps for which filtering should be erased.

downfilter (*t=None*)

Function erases the timestep *t* from the class dictionaries (Measurement, X, Prediction)

Parameters *t* (*int, optional*) – Time-steps for which filtering should be erased.

fit (*u, z*)

Function to auto-evaluate all measurements *z* with control-vectors *u* and starting probability *p*. It returns the believed values *x*, the corresponding probabilities *p* and the predictions *x_tilde*.

Parameters

- **u** (*array_like*) – List of control-vectors.
- **z** (*list of PenguTrack.Measurement objects*) – Recent measurements.

initial_update (*z, i*)

Function to get updates to the corresponding model in the first time step. Handles time-stamps and measurement-vectors. This function also handles the assignment of all incoming measurements to the active sub-filters.

Parameters

- **z** (*list of PenguTrack.Measurement objects*) – Recent measurements.
- **i** (*int*) – Recent/corresponding time-stamp.

Returns

- **z** (*list of PenguTrack.Measurement objects*) – Recent measurements.
- **i** (*int*) – Recent/corresponding time-stamp.

log_prob (***kwargs*)

Function to calculate the probability measure by predictions, measurements and corresponding distributions.

Parameters **keys** (*list of int, optional*) – Time-steps for which probability should be calculated.

Returns **prob** (*float*) – Probability of measurements at the given time-keys.

predict (*u=None, i=None*)

Function to get predictions from the corresponding sub-filter models. Handles time-stamps and control-vectors.

Parameters

- **u** (*array_like, optional*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

unpredict (*t=None*)

Function erases the time-step *t* from the Believes, Predictions and Controls.

Parameters **t** (*int, optional*) – Time-steps for which filtering should be erased.

update (*z=None, i=None, big_jumps=False, verbose=False*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors. This function also handles the assignment of all incoming measurements to the active sub-filters.

Parameters

- **z** (*list of PenguTrack.Measurement objects*) – Recent measurements.
- **i** (*int*) – Recent/corresponding time-stamp.

Returns

- **z** (*list of PenguTrack.Measurement objects*) – Recent measurements.
- **i** (*int*) – Recent/corresponding time-stamp.

class PenguTrack.Filters.**NetworkTracker** (**args, **kwargs*)

This Class describes a filter, which is capable of assigning measurements to tracks, which again are represented by sub-filters. The type of these can be specified, as well as a physical model for predictions. With these objects it is possible to assign possibilities to combinations of measurement and prediction.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.

- **Filter_Class** (*PenguTrack.Filter object*) – A Type of Filter from which all subfilters should be built.
- **Filters** (*dict*) – Dictionary containing all sub-filters as *PenguTrack.Filter* objects.
- **Active_Filters** (*dict*) – Dictionary containing all sub-filters, which are currently updated.
- **FilterThreshold** (*int*) – Number of time steps, before a filter is set inactive.
- **LogProbabilityThreshold** (*float*) – Threshold, under which log-probabilities are concerned negligible.
- **filter_args** (*list*) – Filter-Type specific arguments from the Multi-Filter initialisation can be stored here.
- **filter_kwargs** (*dict*) – Filter-Type specific keyword-arguments from the Multi-Filter initialisation can be stored here.
- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.
- **Order** (*int*) – The order of network assignments. The higher, the more costly and precise the assignment will be.

```
class PenguTrack.Filters.ParticleFilter(model,      n=100,      meas_dist=<MagicMock
                                         name='mock.stats.uniform()'
                                         id='139792175666512'>, state_dist=<MagicMock
                                         name='mock.stats.uniform()'
                                         id='139792175666512'>)
```

This Class describes a particle-filter in the pengu-track package. It calculates actual believed values from predictions and measurements.

Variables

- **Model** (*PenguTrack.model object*) – A physical model to gain predictions from data.
- **N** (*int*) – The number of particles.
- **Particles** (*dict*) – The current particle state-vectors.
- **Weights** (*dict*) – Weights for every particle. Calculated from probability.

- **Measurement_Distribution** (*scipy.stats.distributions object*) – The distribution which describes measurement uncertainty.
- **State_Distribution** (*scipy.stats.distributions object*) – The distribution which describes state vector fluctuations.
- **X** (*dict*) – The time series of believes calculated for this filter. The keys equal the time stamp.
- **X_error** (*dict*) – The time series of errors on the corresponding believes. The keys equal the time stamp.
- **Predicted_X** (*dict*) – The time series of predictions made from the associated data. The keys equal the time stamp.
- **Predicted_X_error** (*dict*) – The time series of estimated prediction errors. The keys equal the time stamp.
- **Measurements** (*dict*) – The time series of measurements assigned to this filter. The keys equal the time stamp.
- **Controls** (*dict*) – The time series of control-vectors assigned to this filter. The keys equal the time stamp.

predict (*u=None, i=None*)

Function to get predictions from the corresponding model. Handles time-stamps and control-vectors.

Parameters

- **u** (*array_like, optional*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **u** (*array_like*) – Recent control-vector.
- **i** (*int*) – Recent/corresponding time-stamp.

update (*z=None, i=None*)

Function to get updates to the corresponding model. Handles time-stamps and measurement-vectors.

Parameters

- **z** (*PenguTrack.Measurement, optional*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp

Returns

- **z** (*PenguTrack.Measurement*) – Recent measurement.
- **i** (*int*) – Recent/corresponding time-stamp.

class PenguTrack.Filters.**Tracker** (*_filter, model, *args, **kwargs*)

2.3 Models

Module containing model classes to be used with pengu-track detectors and filters.

class `PenguTrack.Models.AR(*args, **kwargs)`

This Class describes an auto-regressive model.

Variables

- **Order** (*int*) – Order of the AR-Process. Order = 1 equals an AR1-Process.
- **Coefficients** (*np.ndarray*) – Coefficients of the AR-Process. These describe the time-dependent behaviour of the model.
- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.ndarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.ndarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.ndarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.ndarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

class `PenguTrack.Models.Ballistic(*args, **kwargs)`

This Class describes an simple ballistic model.

Variables

- **Damping** (*np.ndarray*) – Damping constant(s) for ballistic model.
- **Mass** (*float*) – Mass of Object.
- **Timeconst** (*float*) – Step-width of time-discretization.
- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.

- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.asarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.asarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.asarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.asarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

class PenguTrack.Models.BallisticWSpeed (*args, **kwargs)

This Class describes a simple model, assuming slow changing speed-vectors.

Variables

- **Damping** (*np.asarray*) – Damping constant(s) for ballistic model.
- **Timeconst** (*float*) – Step-width of time-discretization.
- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings MultiKal.Model.optimize(state_dict[0])*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.asarray*) – The evolution-matrix of the unperturbed system.

- **Control_Matrix** (*np.ndarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.ndarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.ndarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

class PenguTrack.Models.**MA** (*args, **kwargs)

This Class describes an moving-average model.

Variables

- **Order** (*int*) – Order of the MA-Process. Order = 1 equals an MA1-Process.
- **Coefficients** (*np.ndarray*) – Coefficients of the AR-Process. These describe the time-dependent behaviour of the model.
- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.ndarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.ndarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.ndarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.ndarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

class PenguTrack.Models.**Model** (*args, **kwargs)

This Class describes the abstract function of a physical model in the pengu-track package. It is only meant for subclassing.

Variables

- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.asarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.asarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.asarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.asarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

evolute (*random_vector, state_vector=None*)

Function to predict next measurement from current state.

Parameters

- **random_vector** (*array_like*) – Vector containing the statistical fluctuations.
- **state_vector** (*array_like, optional*) – Latest state vector.

Returns *state* (*np.asarray*) – Calculated state vector.

infer_state (*meas_vector*)

Tries to infer state from measurement.

Parameters *meas_vector* (*array_like*) – Vector containing the measurement.

Returns *state* (*np.asarray*) – Calculated state vector.

Raises **LinAlgError** – If the state can not be inferred do to singularity in a matrix-inversion.

measure (*state_vector*)

Function to predict next measurement from current state.

Parameters *state_vector* (*array_like*) – Latest state vector.

Returns *measurement* (*np.asarray*) – Expected measurement vector.

predict (*state_vector*, *control_vector*)

Function to predict next state from current state and external control.

Parameters

- **state_vector** (*array_like*) – Latest state vector.
- **control_vector** (*array_like*) – Latest control vector.

Returns **prediction** (*np.ndarray*) – New state vector.

static pseudo_inverse (*matrix*)

Calculates an alternative inverse for non square (non invertible) matrices.

Parameters **matrix** (*array_like*) – Non square Matrix to be inverted.

Returns **pseudo-inverse** (*np.ndarray*) – Calculated pseudo-inverse.

class `PenguTrack.Models.RandomWalk` (**args*, ***kwargs*)

This Class describes an easy random walk model.

Variables

- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.ndarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.ndarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.ndarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.ndarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

class `PenguTrack.Models.VariableSpeed` (**args*, ***kwargs*)

This Class describes a simple model, assuming slow changing speed-vectors.

Variables

- **Damping** (*np.ndarray*) – Damping constant(s) for ballistic model.

- **Timeconst** (*float*) – Step-width of time-discretization.
- **State_dim** (*int*) – Number of entries in the state-vector.
- **Control_dim** (*int*) – Number of entries in the control-vector.
- **Meas_dim** (*int*) – Number of entries in the measurement-vector.
- **Evolution_dim** (*int*) – Number of entries in the evolution-vector.
- **Opt_Params** (*list of strings MultiKal.Model.optimize(state_dict[0])*) – Parameters of model, which can be optimized.
- **Opt_Params_Shape** (*dict*) – Keys are Opt_Params, entries are tuples containing shapes of the corresponding Parameters.
- **Opt_Params_Borders** (*dict*) – Keys are Opt_Params, entries are tuples containing borders of the corresponding Parameters.
- **Initial_Args** (*list*) – The arguments, which were given in the init function.
- **Initial_KWArgs** (*dict*) – The keyword-arguments, which were given in the init function.
- **State_Matrix** (*np.asarray*) – The evolution-matrix of the unperturbed system.
- **Control_Matrix** (*np.asarray*) – The evolution-matrix, which shows the influence of external control.
- **Measurement_Matrix** (*np.asarray*) – The matrix, which shows how the state vectors are projected into a measurement vector.
- **Evolution_Matrix** (*np.asarray*) – The matrix, which shows the influence of statistical fluctuations to the state.
- **Measured_Variables** (*list*) – List of variables, that are measured within the model.
- **State_Variables** (*list*) – List of variables, that are tracked within the model.
- **Extensions** (*list*) – List of measured parameters not included in standard model.

2.4 Stitches

class PenguTrack.Stitchers.**Heublein_Stitcher** (*a1, a2, a3, a4, a5, max_cost, limiter*)

Class for Stitching Tracks

Delete_Tracks_with_length_1 ()

Deletes tracks with a length of 1.

average_Measurements (*meas1, meas2*)

returns mean of two measurements

find_matches (*Tracks, ids*)

Uses the points function to find the best matches.

Parameters

Tracks: **list** list of tracks in the right format.

ids: **list** list of track ids.

getHeubleinTracks ()

Returns the tracks in the right format.

stitch()

Uses all the functions and runs the main process

PYTHON MODULE INDEX

p

PenguTrack.Detectors, [5](#)

PenguTrack.Filters, [11](#)

PenguTrack.Models, [22](#)

PenguTrack.Stitchers, [27](#)

A

AdaptedKalmanFilter (class in *PenguTrack.Filters*), 11
 AdvancedKalmanFilter (class in *PenguTrack.Filters*), 11
 AlexSegmentation (class in *PenguTrack.Detectors*), 5
 AR (class in *PenguTrack.Models*), 22
 AreaBlobDetector (class in *PenguTrack.Detectors*), 5
 AreaDetector (class in *PenguTrack.Detectors*), 5
 average_Measurements () (*PenguTrack.Stitchers.Heublein_Stitcher* method), 27

B

Ballistic (class in *PenguTrack.Models*), 22
 BallisticWSpeed (class in *PenguTrack.Models*), 23
 BlobDetector (class in *PenguTrack.Detectors*), 5
 BlobSegmentation (class in *PenguTrack.Detectors*), 6

D

Delete_Tracks_with_length_1 () (*PenguTrack.Stitchers.Heublein_Stitcher* method), 27
 detect () (*PenguTrack.Detectors.AreaBlobDetector* method), 5
 detect () (*PenguTrack.Detectors.AreaDetector* method), 5
 detect () (*PenguTrack.Detectors.BlobDetector* method), 6
 detect () (*PenguTrack.Detectors.BlobSegmentation* method), 6
 detect () (*PenguTrack.Detectors.MoGSegmentation* method), 6
 detect () (*PenguTrack.Detectors.MoGSegmentation2* method), 7
 detect () (*PenguTrack.Detectors.SimpleAreaDetector* method), 7
 detect () (*PenguTrack.Detectors.SimpleAreaDetector2* method), 7

detect () (*PenguTrack.Detectors.TinaCellDetector* method), 8
 detect () (*PenguTrack.Detectors.ViBeSegmentation* method), 8
 detect () (*PenguTrack.Detectors.WatershedDetector* method), 8
 Detector (class in *PenguTrack.Detectors*), 6
 dotdict (class in *PenguTrack.Detectors*), 8
 downdate () (*PenguTrack.Filters.Filter* method), 13
 downdate () (*PenguTrack.Filters.MultiFilter* method), 18
 downfilter () (*PenguTrack.Filters.Filter* method), 13
 downfilter () (*PenguTrack.Filters.MultiFilter* method), 18
 DumbViBeSegmentation (class in *PenguTrack.Detectors*), 6

E

EmperorDetector (class in *PenguTrack.Detectors*), 6
 evolve () (*PenguTrack.Models.Model* method), 25
 extended_regionprops () (in module *PenguTrack.Detectors*), 8

F

Filter (class in *PenguTrack.Filters*), 12
 filter () (*PenguTrack.Filters.Filter* method), 13
 find_matches () (*PenguTrack.Stitchers.Heublein_Stitcher* method), 27
 fit () (*PenguTrack.Filters.Filter* method), 13
 fit () (*PenguTrack.Filters.MultiFilter* method), 18
 FlowDetector (class in *PenguTrack.Detectors*), 6

G

getHeubleinTracks () (*PenguTrack.Stitchers.Heublein_Stitcher* method), 27

H

Heublein_Stitcher (class in *PenguTrack.Stitchers*), 27
 HungarianTracker (class in *PenguTrack.Filters*), 14
 HybridSolver (class in *PenguTrack.Filters*), 14

I

`infer_state()` (*PenguTrack.Models.Model* method), 25
`InformationFilter` (class in *PenguTrack.Filters*), 15
`initial_update()` (*PenguTrack.Filters.MultiFilter* method), 18

K

`KalmanBaseFilter` (class in *PenguTrack.Filters*), 16
`KalmanFilter` (class in *PenguTrack.Filters*), 17

L

`log_prob()` (*PenguTrack.Filters.MultiFilter* method), 19

M

`MA` (class in *PenguTrack.Models*), 24
`MeanViBeSegmentation` (class in *PenguTrack.Detectors*), 6
`measure()` (*PenguTrack.Models.Model* method), 25
`Measurement` (class in *PenguTrack.Detectors*), 6
`Model` (class in *PenguTrack.Models*), 24
`MoGSegmentation` (class in *PenguTrack.Detectors*), 6
`MoGSegmentation2` (class in *PenguTrack.Detectors*), 7
`MultiFilter` (class in *PenguTrack.Filters*), 17

N

`NetworkTracker` (class in *PenguTrack.Filters*), 19
`NKCellDetector` (class in *PenguTrack.Detectors*), 7
`NKCellDetector2` (class in *PenguTrack.Detectors*), 7

P

`ParticleFilter` (class in *PenguTrack.Filters*), 20
`PenguTrack.Detectors` (module), 5
`PenguTrack.Filters` (module), 11
`PenguTrack.Models` (module), 22
`PenguTrack.Stitchers` (module), 27
`predict()` (*PenguTrack.Filters.AdaptedKalmanFilter* method), 11
`predict()` (*PenguTrack.Filters.Filter* method), 13
`predict()` (*PenguTrack.Filters.InformationFilter* method), 16
`predict()` (*PenguTrack.Filters.KalmanFilter* method), 17
`predict()` (*PenguTrack.Filters.MultiFilter* method), 19
`predict()` (*PenguTrack.Filters.ParticleFilter* method), 21
`predict()` (*PenguTrack.Models.Model* method), 25
`pseudo_inverse()` (*PenguTrack.Models.Model* static method), 26

R

`RandomWalk` (class in *PenguTrack.Models*), 26
`RegionPropDetector` (class in *PenguTrack.Detectors*), 7

S

`segmentate()` (*PenguTrack.Detectors.MoGSegmentation* method), 6
`Segmentation` (class in *PenguTrack.Detectors*), 7
`SimpleAreaDetector` (class in *PenguTrack.Detectors*), 7
`SimpleAreaDetector2` (class in *PenguTrack.Detectors*), 7
`stitch()` (*PenguTrack.Stitchers.Heublein_Stitcher* method), 27

T

`TCellDetector` (class in *PenguTrack.Detectors*), 7
`ThresholdSegmentation` (class in *PenguTrack.Detectors*), 7
`TinaCellDetector` (class in *PenguTrack.Detectors*), 7
`Tracker` (class in *PenguTrack.Filters*), 21

U

`unpredict()` (*PenguTrack.Filters.Filter* method), 14
`unpredict()` (*PenguTrack.Filters.MultiFilter* method), 19
`update()` (*PenguTrack.Filters.AdaptedKalmanFilter* method), 11
`update()` (*PenguTrack.Filters.AdvancedKalmanFilter* method), 12
`update()` (*PenguTrack.Filters.Filter* method), 14
`update()` (*PenguTrack.Filters.InformationFilter* method), 16
`update()` (*PenguTrack.Filters.KalmanFilter* method), 17
`update()` (*PenguTrack.Filters.MultiFilter* method), 19
`update()` (*PenguTrack.Filters.ParticleFilter* method), 21

V

`VariableSpeed` (class in *PenguTrack.Models*), 26
`VarianceSegmentation` (class in *PenguTrack.Detectors*), 8
`ViBeSegmentation` (class in *PenguTrack.Detectors*), 8

W

`WatershedDetector` (class in *PenguTrack.Detectors*), 8