In[22]:=

```
ClearAll["Global`*"]
SetOptions[$FrontEndSession, NotebookAutoSave → True]
NotebookSave[]
```

Lynx paper notebook 1

Script for analysis presented with respect to hoverfly flights

Preprocessing

Load Trajectory3D package

Package available at Github, download and move to the Applications subdirectory of your user base directory.

In[ ]:=

```
$UserBaseDirectory
```

In[35]:=

```
Needs["Trajectory3D`"]
Names["Trajectory3D`*"]
```

Out[36]=

```
{AngleofFlight, CollettPlot3D, DistanceProfile3D, GetData3D,
 InputUserValues3D, OrthogonalComponentsVelocity3D, ProximityCut3D,
 SpeedCollettPlot3D, SpeedProfile3D, TwinCollettPlot3D}
```

Import Data

project specific constants and other variables

In[27]:=

```
fps = 1 / 500;(*frame rate*)
flyfolder =
  "/Users/dinesh/Dropbox/projects/lynx/lynx prey response/Data/processed
    data/flydata"; (*insert folder path to csv files *)

data = Import[#, "CSV"] & /@ FileNames["*.csv", flyfolder];
```

Categorize data

In[30]:=
```
flights = Range@Length@data
```

Out[30]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27}
```

Assign flights to flower type

In[31]:=
```
fyellows = {2, 3, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25};
```

In[32]:=
```
fwhites = {1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 26, 27};
```

Segment trajectory to closest approach to flower (point chosen visually)

In[33]:=
```
SpecialProximityCut3D[file_, n_] := Module[{head, obj, cuts, cuttraj},
head = file⟦All, {1, 2, 3}⟧;
obj = file⟦All, {7, 8, 9}⟧;
cuts = {75, 282, 119, 90, 58, 285, 328, 242, 471, 311, 117, 85, 404,
    257, 75, 497, 65, 328, 245, 250, 118, 53, 54, 182, 258, 10, 19};
cuttraj = file⟦ ;; cuts⟦n⟧, All⟧;
cuttraj
]
```

In[34]:=
```
mpc = SpecialProximityCut3D[data⟦#⟧, #] & /@ flights;
```

Analyses

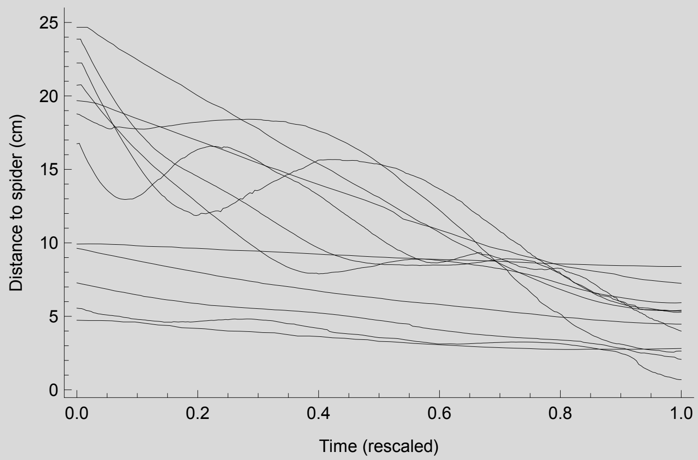
Distance

Distance profile

In[ ]:=

```
ListLinePlot[
 N@TimeSeriesRescale[DistanceProfile3D[mpc[[#]]], {0, 1}] & /@ fyellows,
 PlotStyle → Directive[{Black, Thin}], Frame → {{True, False}, {True, False}},
 FrameLabel → {{HoldForm["Distance to spider (cm)"], None},
   {HoldForm["Time (rescaled)"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

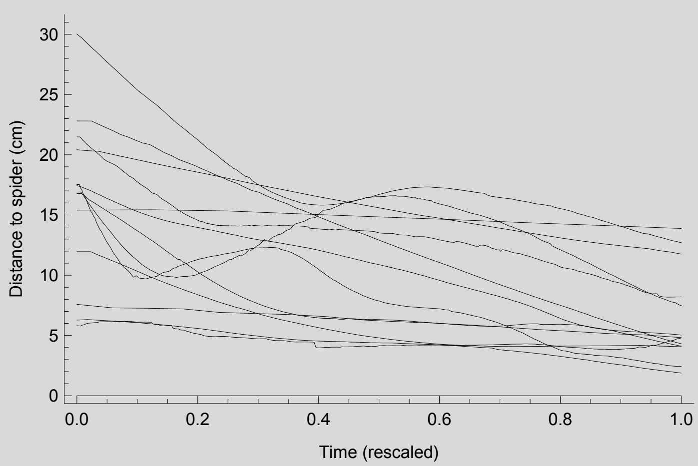
Out[ ]=



In[ ]:=

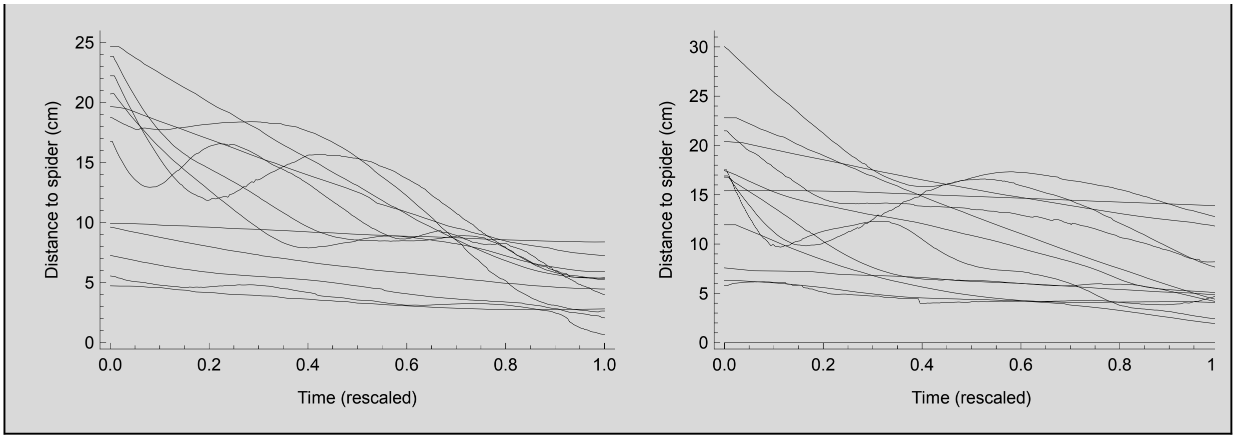
```
ListLinePlot[
 N@TimeSeriesRescale[DistanceProfile3D[mpc[[#]]], {0, 1}] & /@ fwhites[[2 ;;]],
 PlotStyle → Directive[{Black, Thin}], Frame → {{True, False}, {True, False}},
 FrameLabel → {{HoldForm["Distance to spider (cm)"], None},
   {HoldForm["Time (rescaled)"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}, PlotRange → All]
```

Out[ ]=



for fig

In[ ]:=
```
GraphicsRow[{ , }]
```
Out[ ]=



minimum distance Yellows vs whites

In[ ]:=
```
minyellow = Min[DistanceProfile3D[mpc[[#]]]] & /@ fyellows
```
Out[ ]=
```
{5.91197, 4.46833, 3.99423, 7.24363, 5.35963, 2.55969,
 5.27143, 5.38142, 8.38861, 2.74377, 2.06915, 0.689805}
```

In[ ]:=
```
minwhites = Min[DistanceProfile3D[mpc[[#]]]] & /@ fwhites
```
Out[ ]=
```
{25.562, 4.0714, 11.7528, 5.03334, 3.83508, 4.11694, 2.42063, 7.48262,
 1.87721, 4.3131, 8.17715, 12.7003, 3.71957 × 10^-12, 13.8802, 4.82305}
```

In[ ]:=
```
BoxWhiskerChart[{minwhites, minyellow},
 ChartLabels → {"White Flower", "Yellow Flower"}, FrameLabel →
  {{HoldForm["Closest distance to spider (cm)"], None}, {None, None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```
Out[ ]=

Test whether the minimum distances are significantly different

In[◦]:=
```
Mean /@ {minwhites, minyellow}
```
Out[◦]=
```
{7.33639, 4.50681}
```

In[◦]:=
```
StandardDeviation /@ {minwhites, minyellow}
```
Out[◦]=
```
{6.46878, 2.2215}
```

In[◦]:=
```
MannWhitneyTest[{minwhites, minyellow}, Automatic, "TestDataTable"]
```
Out[◦]=

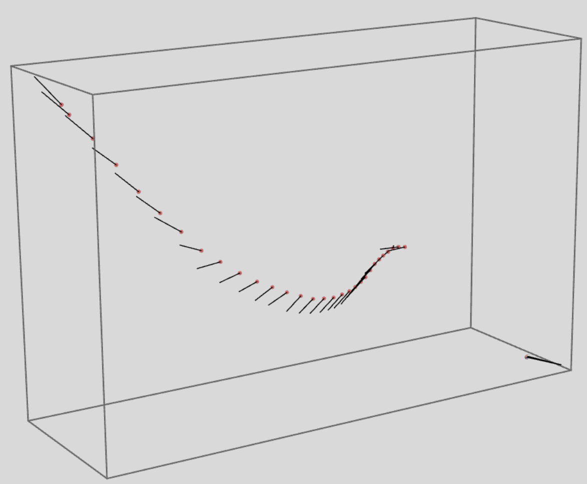
	Statistic	P-Value
Mann- Whitney	76.	0.48562

Trajectory Plots

Plots of selected trajectories

Ball and pin plot of a typical trajectory

In[◦]:=
```
TwinCollettPlot3D[mpc〚4〛]
```
Out[◦]=



In[•]:=
```
Show[, ViewPoint → {1.3, -2.4, 2.}, Boxed → False]
```

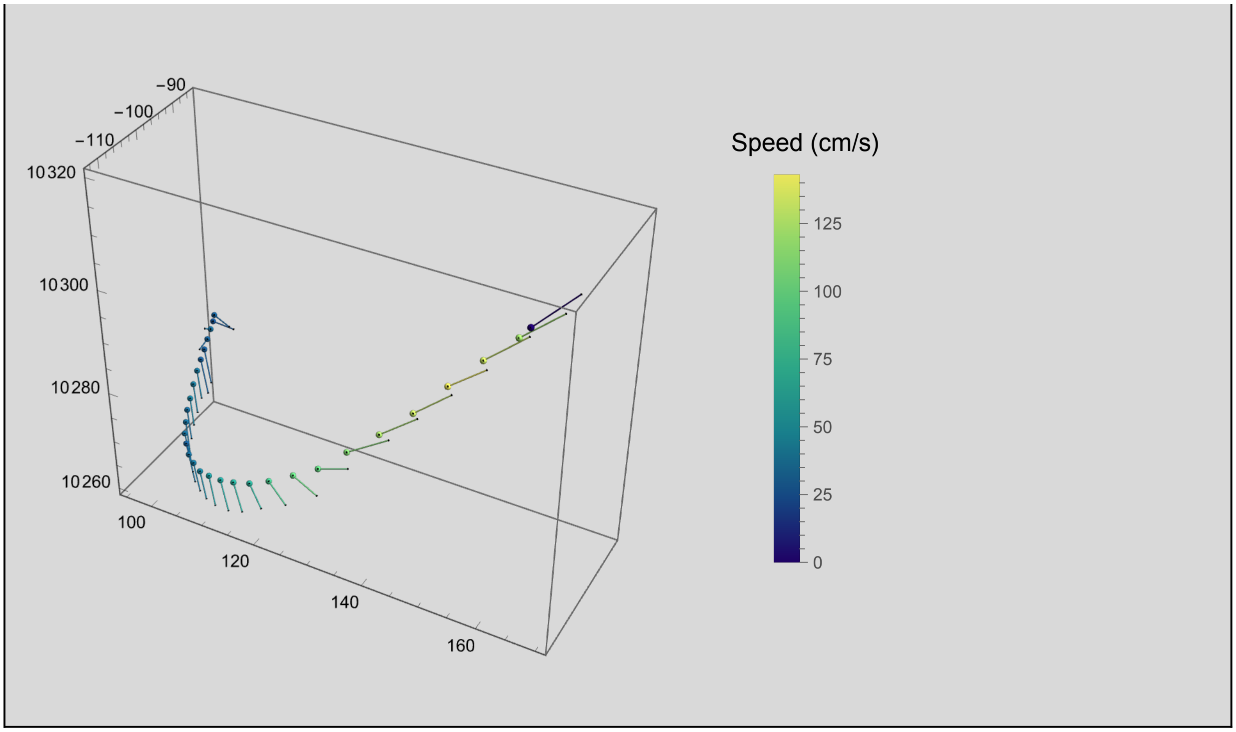
Out[•]=

All trajectories (not run)

```
TwinCollettPlot3D[mpc[[#]]] & /@ flights;
```

Colour coded to speed

In[ ]:=
```
SpeedCollettPlot3D[mpc⟦4⟧]
```
Out[ ]=



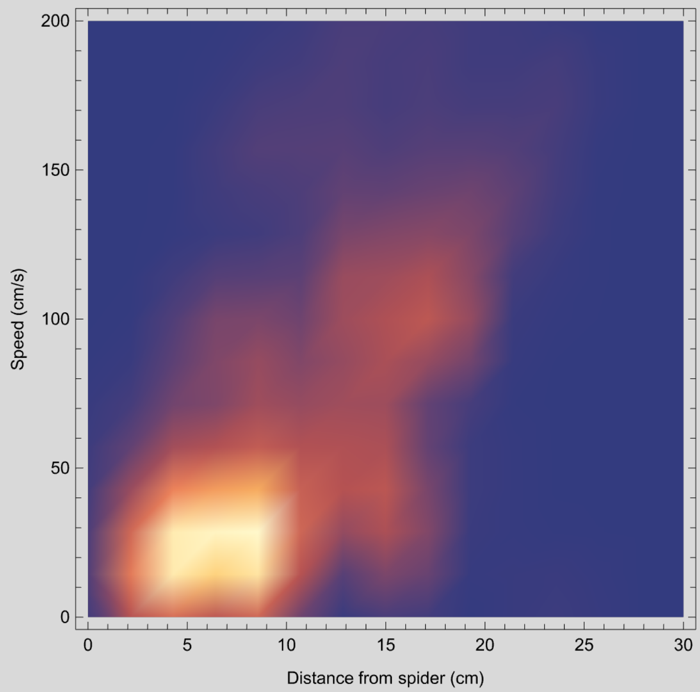
Distance vs Speed

In[37]:=
```
yellowdistspeeds =
  {DistanceProfile3D[mpc⟦#⟧], SpeedProfile3D[mpc⟦#⟧]}ᵀ & /@ fyellows;
```

In[38]:=
```
whitedistspeeds =
  {DistanceProfile3D[mpc⟦#⟧], SpeedProfile3D[mpc⟦#⟧]}ᵀ & /@ fwhites;
```

In[ ]:=
```
SmoothDensityHistogram[Flatten[yellowdistspeeds, 1],
 PlotRange → {{0, 30}, {0, 200}}, FrameLabel → {{HoldForm["Speed (cm/s)"], None},
   {HoldForm["Distance from spider (cm)"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Out[ ]=



Test for significance

In[50]:=

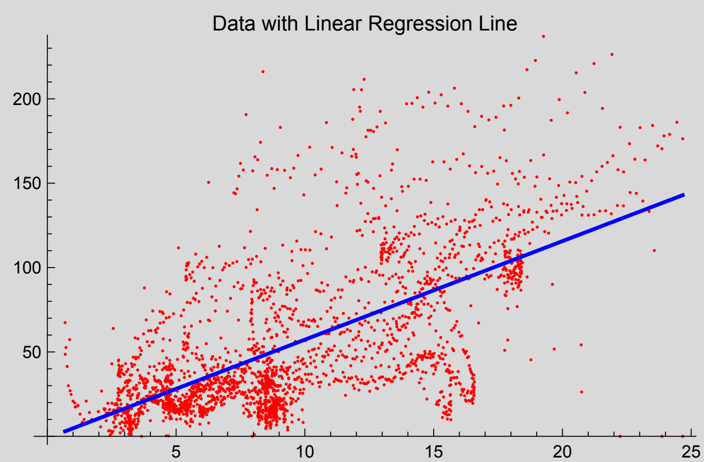
```
(* Assuming yellowdistspeeds data is structured as {x, y} pairs *)
ydata = Flatten[yellowdistspeeds, 1];

(* Fit a linear model *)
ymodel = LinearModelFit[ydata, x, x];

(* Show the plot with data and regression line *)
Show[
  ListPlot[ydata, PlotStyle → Red,
   PlotLabel → "Data with Linear Regression Line"],
  Plot[ymodel[x], {x, Min[ydata[[All, 1]]], Max[ydata[[All, 1]]]}, PlotStyle → Blue]
]

(* Extract parameter significance *)
parameterTable = ymodel["ParameterTable"];
parameterTable
```

Out[52]=



Out[54]=

	Estimate	Standard Error	t-Statistic	P-Value
1	−1.08304	1.60672	−0.674068	0.500332
x	5.83697	0.147782	39.497	3.80785×10^{-264}

In[55]:=

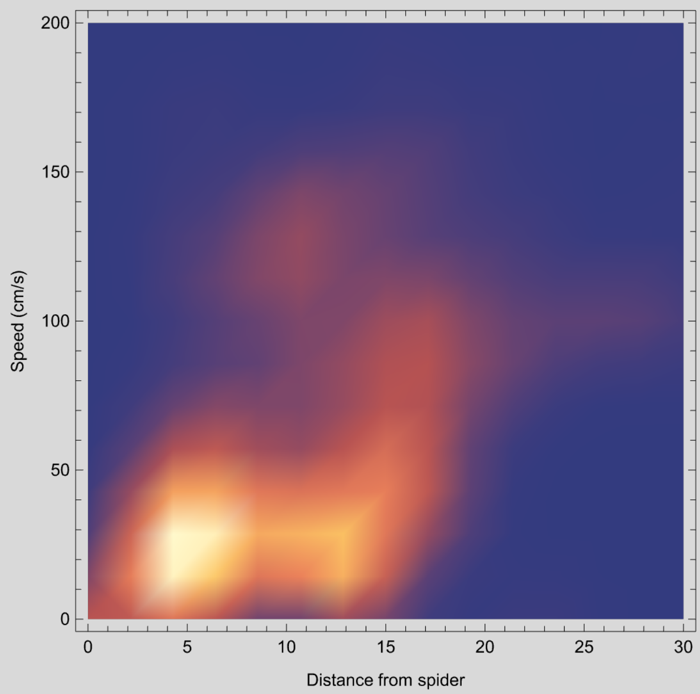
```
Grid[Transpose[{#, ymodel[#]} &[{"AdjustedRSquared", "AIC", "BIC", "RSquared"}]],
 Alignment → Left]
```

Out[55]=

AdjustedRSquared	0.390045
AIC	24 475.6
BIC	24 493.
RSquared	0.390295

In[∘]:=
```
SmoothDensityHistogram[Flatten[whitedistspeeds, 1],
 PlotRange → {{0, 30}, {0, 200}}, FrameLabel →
  {{HoldForm["Speed (cm/s)"], None}, {HoldForm["Distance from spider"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Out[∘]=



Test for significance

In[39]:=

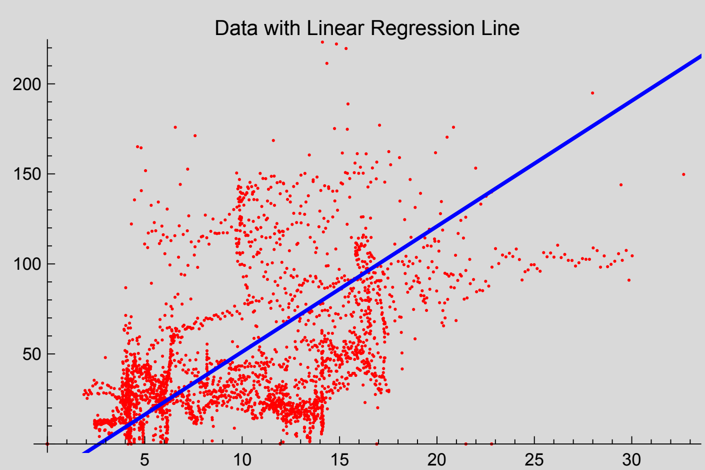
```
(* Assuming yellowdistspeeds data is structured as {x, y} pairs *)
wdata = Flatten[whitedistspeeds, 1];

(* Fit a linear model *)
wmodel = LinearModelFit[wdata, x, x];

(* Show the plot with data and regression line *)
Show[
  ListPlot[wdata, PlotStyle → Red,
   PlotLabel → "Data with Linear Regression Line"],
  Plot[wmodel[x], {x, Min[wdata[[All, 1]]], Max[wdata[[All, 1]]]}, PlotStyle → Blue]
]

(* Extract parameter significance *)
parameterTable = wmodel["ParameterTable"];
parameterTable
```

Out[41]=



General : Exp[−1064.36] is too small to represent as a normalized machine number; precision may be lost.

Out[43]=

	Estimate	Standard Error	t-Statistic	P-Value
1	−18.5431	1.59516	−11.6246	1.52749×10^{-30}
x	6.97139	0.123872	56.2792	0.

In[44]:=

```
Grid[Transpose[{#, wmodel[#]} &[{"AdjustedRSquared", "AIC", "BIC", "RSquared"}]],
 Alignment → Left]
```

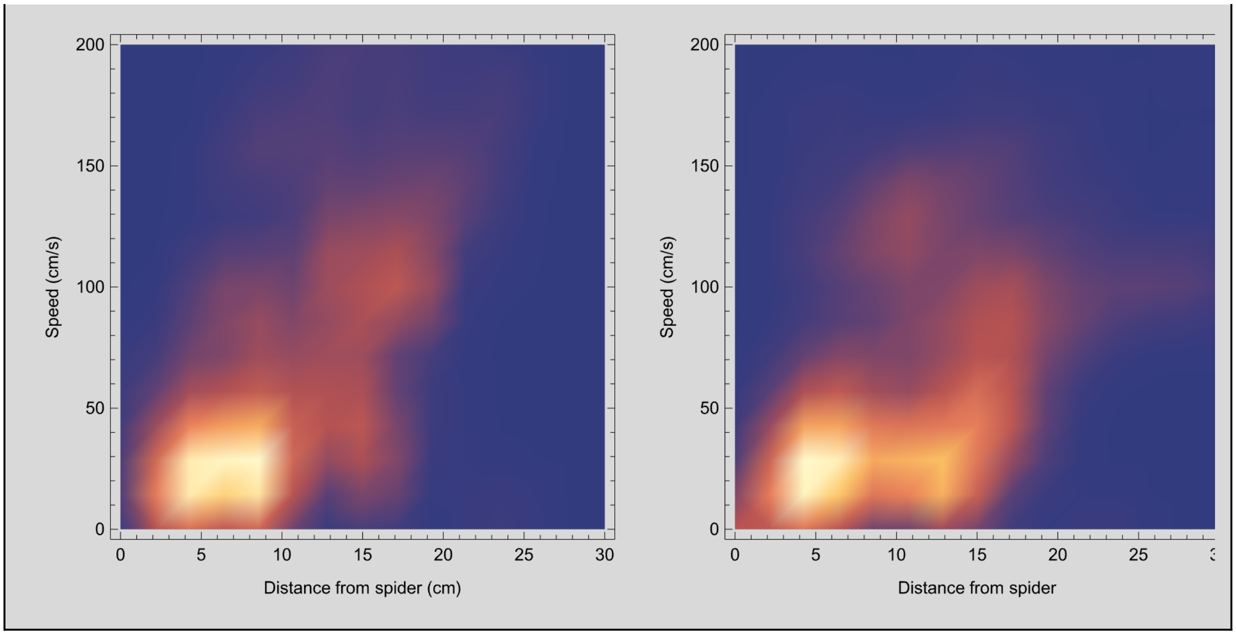
Out[44]=

AdjustedRSquared	0.529725
AIC	29525.2
BIC	29543.
RSquared	0.529892

For figure

In[ ]:=
```
GraphicsRow[{14, }]
```

Out[ ]=



In[ ]:=
```
Rasterize[%59, "Image"]
```

In[ ]:=
```
Export["/Users/dinesh/Dropbox/projects/lynx/lynx prey
    response/Manuscript/Figures/distvsspeed_fly.pdf", %64, "PDF"]
```

Persistence Velocity

Yellow flower flights

Extract persistent velocity values for yellow flower flights

In[ ]:=
```
PVyellows = OrthogonalComponentsVelocity3D[mpc[[#]]][[1]] & /@ fyellows;
```

In[ ]:=
```
ypv = LowpassFilter[PVyellows[[#]], 0.5] & /@ Range@Length@fyellows;
```

Distances for yellow flower flights

In[ ]:=
```
ydis = DistanceProfile3D[mpc[[#]]] & /@ fyellows;
```

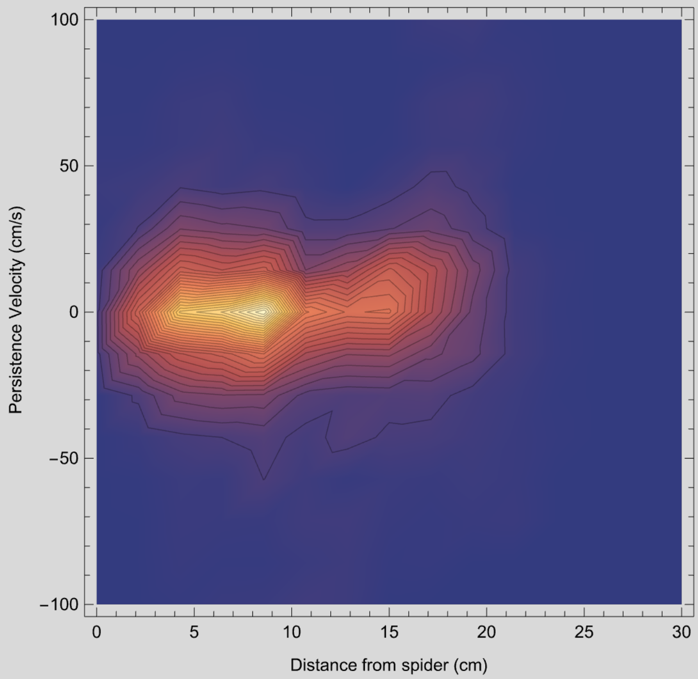
Join to a single dataset

In[ ]:=
```
yellowdistPV = {ydis[[#]][[2 ;;]], ypv[[#]]}ᵀ & /@ {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
```

Smooth density histogram with contours

In[∘]:=
```
SmoothDensityHistogram[Flatten[yellowdistPV, 1],
 PlotRange → {{0, 30}, {-100, 100}}, Mesh → 30,
 FrameLabel → {{HoldForm[" Persistence Velocity (cm/s)"], None},
   {HoldForm["Distance from spider (cm)"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Out[∘]=



White flower flights

Extract persistent velocity values for white flower flights

In[∘]:=
```
PVwhites = OrthogonalComponentsVelocity3D[mpc〚#〛]〚1〛 & /@ fwhites;
```

Subsample to remove trajectories where computation did not work; standardize (z-score normalisation) and then run a low pass filter

In[∘]:=
```
wpv = LowpassFilter[PVwhites〚#〛, 0.5] & /@ Range@Length@fwhites;
```

Distances for white flower flights

In[∘]:=
```
wdis = DistanceProfile3D[mpc〚#〛] & /@ fwhites;
```

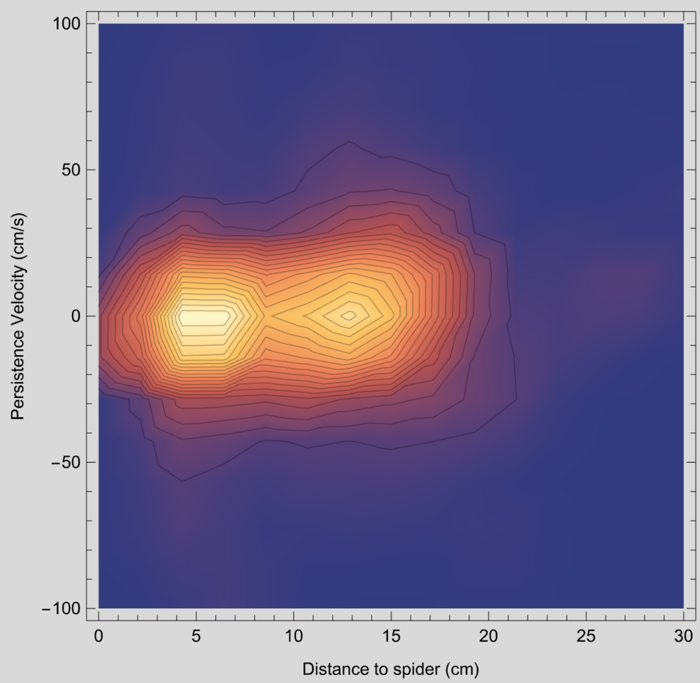
Join to a single dataset

In[∘]:=
```
whitedistPV = {wdis〚#〛〚2 ;;〛, wpv〚#〛}ᵀ & /@ Range@Length@fwhites;
```

Smooth density histogram with contours

In[ ]:=
```
SmoothDensityHistogram[Flatten[whitedistPV, 1],
 PlotRange → {{0, 30}, {-100, 100}}, Mesh → 20,
 FrameLabel → {{HoldForm["Persistence Velocity (cm/s)"], None},
   {HoldForm["Distance to spider (cm)"], None}},
 PlotLabel → None, LabelStyle → {GrayLevel[0]}]
```

Out[ ]=



For figure

In[ ]:=

```
GraphicsRow[{20, }]
```

Out[ ]=