3d Visualizations

I have also found several some other 3d visualizations to be very helpful. First I'll go through the data processing that I used to get them, then show some pretty pictures.

Functions used here: plotSVD: a function that plots several common PCA visualizations RobustPCA: an algorithm that will be explained below; implementation on MathWorks plot_colored: a function for plotting 3d data in different colors

Original paper: Candès, E.J., Li, X., Ma, Y. and Wright, J., 2011. Robust principal component analysis?. Journal of the ACM (JACM), 58(3), p.11.

Robust PCA

This algorithm is not very similar to normal PCA, and I think is better described as a pre-processing step to determine which parts of the data are amenible to PCA or other algorithms. Fundamentally it attempts to decompose a data matrix *X* into: $X = L + S$ where *L* is Low-rank (i.e. the SVD sigma values drop off very quickly), and *S* is Sparse (i.e. there are few non-zero entries). There is a single important parameter here, the penalty for adding an extra non-zero entry to the matrix *S* ($lambda$).

To illustrate the properties of this algorithm, I'll do two visualizations with two different values of $lambda$. NOTE: the MATLAB implementation I'm using is not particularly fast, so the code blocks with robustPCA will take ~1 minute.

First, let's import the data and filter a bit so that it is cleaner. Concatenate the derivatives on the end of the data for a more accurate picture.

```
filename = '../../Collaborations/Zimmer_data/WildType_adult/simplewt5/
wbdataset.mat';
dat_struct = importdata(filename);
my_filter = @(dat,w) filter(ones(w,1)/w,1,dat);
this_dat = my_filter(dat_struct.traces,3).';

use_deriv = true;
if use_deriv
    this_deriv = my_filter(dat_struct.tracesDif,3).';
    this_dat = [this_dat(:,10:end); this_deriv(:,9:end)];
end
```

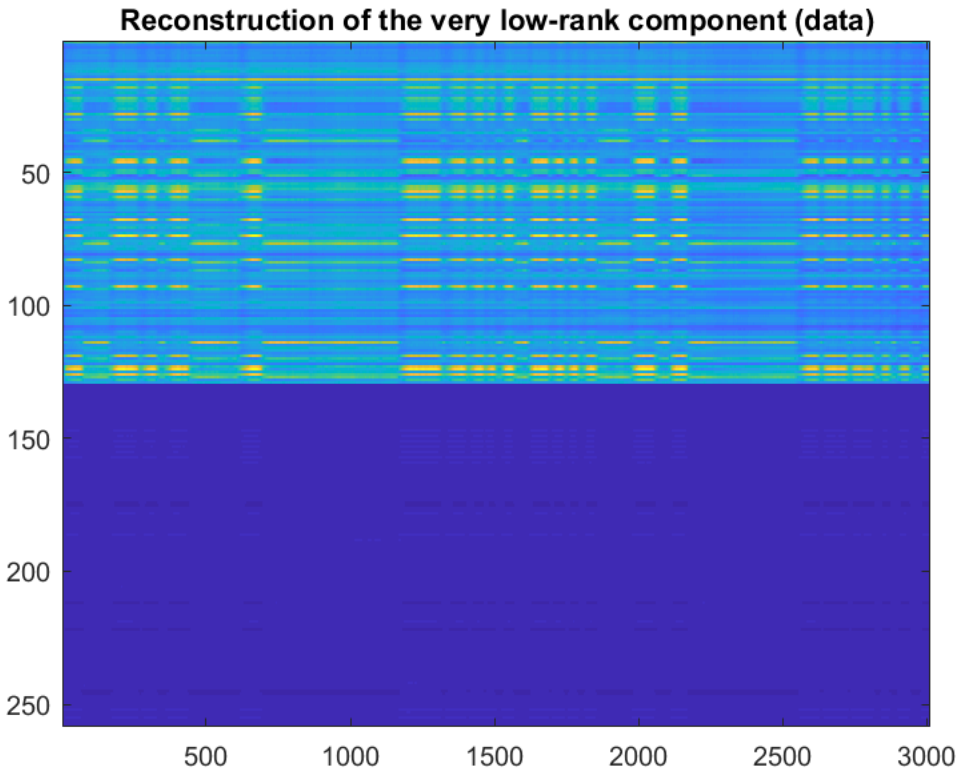
Do the robustPCA algorithm with a small value of lambda. This means that the low-rank component is VERY low-rank.

```
lambda = 0.0065;
[L_very_low, ~] = RobustPCA(this_dat.', lambda);
% Plot the VERY low-rank component
L_filter_very_low = my_filter(L_very_low, 10);
figure;
imagesc(L_filter_very_low.');
title('Reconstruction of the very low-rank component (data)')
```

```
iter: 0001 err: 0.166256 rank(L): 10 card(S): 59245
iter: 0010 err: 0.018711 rank(L): 3 card(S): 528871
iter: 0020 err: 0.006804 rank(L): 4 card(S): 644241
iter: 0030 err: 0.003623 rank(L): 4 card(S): 685705
```

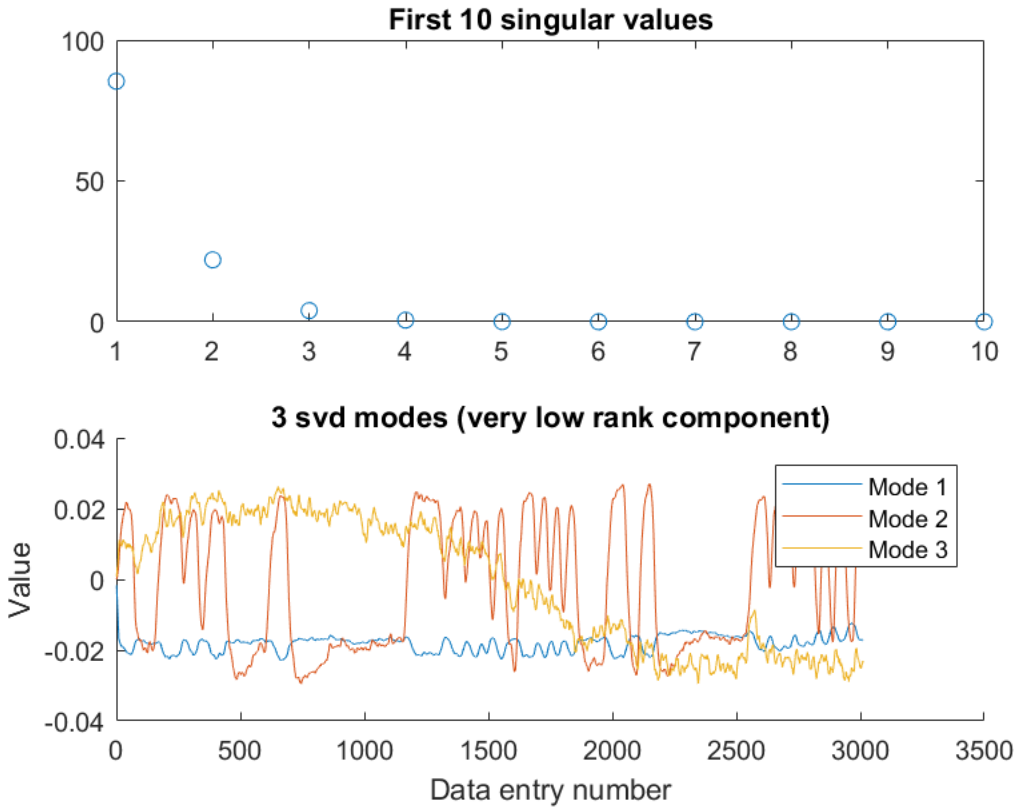
```
iter: 0040 err: 0.002097 rank(L): 4 card(S): 707492
iter: 0050 err: 0.001368 rank(L): 4 card(S): 720454
iter: 0060 err: 0.001005 rank(L): 4 card(S): 729077
iter: 0070 err: 0.000785 rank(L): 4 card(S): 735275
iter: 0080 err: 0.000631 rank(L): 4 card(S): 739939
iter: 0090 err: 0.000522 rank(L): 4 card(S): 743493
iter: 0100 err: 0.000440 rank(L): 4 card(S): 746436
iter: 0110 err: 0.000380 rank(L): 4 card(S): 748721
iter: 0120 err: 0.000330 rank(L): 4 card(S): 750780
iter: 0130 err: 0.000291 rank(L): 4 card(S): 752379
iter: 0140 err: 0.000259 rank(L): 4 card(S): 753793
iter: 0150 err: 0.000233 rank(L): 4 card(S): 754968
iter: 0160 err: 0.000211 rank(L): 4 card(S): 755999
iter: 0170 err: 0.000193 rank(L): 4 card(S): 756885
iter: 0180 err: 0.000177 rank(L): 4 card(S): 757696
iter: 0190 err: 0.000162 rank(L): 4 card(S): 758442
iter: 0200 err: 0.000151 rank(L): 4 card(S): 759057
iter: 0210 err: 0.000139 rank(L): 5 card(S): 759656
iter: 0220 err: 0.000129 rank(L): 5 card(S): 760228
iter: 0230 err: 0.000121 rank(L): 5 card(S): 760677
iter: 0240 err: 0.000114 rank(L): 5 card(S): 761129
iter: 0250 err: 0.000106 rank(L): 5 card(S): 761584
iter: 0260 err: 0.000100 rank(L): 5 card(S): 761976
iter: 0270 err: 0.000094 rank(L): 5 card(S): 762349
iter: 0280 err: 0.000088 rank(L): 5 card(S): 762678
iter: 0290 err: 0.000084 rank(L): 5 card(S): 763004
iter: 0300 err: 0.000079 rank(L): 5 card(S): 763280
iter: 0310 err: 0.000075 rank(L): 5 card(S): 763559
iter: 0320 err: 0.000071 rank(L): 5 card(S): 763805
iter: 0330 err: 0.000068 rank(L): 5 card(S): 764011
iter: 0340 err: 0.000065 rank(L): 5 card(S): 764215
iter: 0350 err: 0.000063 rank(L): 5 card(S): 764429
iter: 0360 err: 0.000060 rank(L): 5 card(S): 764605
iter: 0370 err: 0.000057 rank(L): 5 card(S): 764794
iter: 0380 err: 0.000055 rank(L): 5 card(S): 764953
iter: 0390 err: 0.000053 rank(L): 5 card(S): 765100
iter: 0400 err: 0.000052 rank(L): 5 card(S): 765233
iter: 0410 err: 0.000050 rank(L): 5 card(S): 765377
iter: 0420 err: 0.000048 rank(L): 5 card(S): 765528
iter: 0430 err: 0.000046 rank(L): 5 card(S): 765666
iter: 0440 err: 0.000045 rank(L): 5 card(S): 765785
iter: 0450 err: 0.000043 rank(L): 5 card(S): 765922
iter: 0460 err: 0.000042 rank(L): 5 card(S): 766037
iter: 0470 err: 0.000040 rank(L): 5 card(S): 766151
iter: 0480 err: 0.000039 rank(L): 5 card(S): 766249
iter: 0490 err: 0.000038 rank(L): 5 card(S): 766360
iter: 0500 err: 0.000037 rank(L): 5 card(S): 766460
iter: 0510 err: 0.000036 rank(L): 5 card(S): 766546
iter: 0520 err: 0.000035 rank(L): 5 card(S): 766650
iter: 0530 err: 0.000034 rank(L): 5 card(S): 766734
iter: 0540 err: 0.000033 rank(L): 5 card(S): 766824
iter: 0550 err: 0.000032 rank(L): 5 card(S): 766915
iter: 0560 err: 0.000031 rank(L): 5 card(S): 766993
iter: 0570 err: 0.000030 rank(L): 5 card(S): 767069
```

```
iter: 0580 err: 0.000029 rank(L): 5 card(S): 767141
iter: 0590 err: 0.000029 rank(L): 5 card(S): 767209
iter: 0600 err: 0.000028 rank(L): 5 card(S): 767280
iter: 0610 err: 0.000027 rank(L): 5 card(S): 767360
iter: 0620 err: 0.000026 rank(L): 5 card(S): 767425
iter: 0630 err: 0.000026 rank(L): 5 card(S): 767496
iter: 0640 err: 0.000025 rank(L): 5 card(S): 767558
iter: 0650 err: 0.000024 rank(L): 5 card(S): 767625
iter: 0660 err: 0.000024 rank(L): 5 card(S): 767685
iter: 0670 err: 0.000023 rank(L): 5 card(S): 767751
iter: 0680 err: 0.000022 rank(L): 5 card(S): 767796
iter: 0690 err: 0.000022 rank(L): 5 card(S): 767837
iter: 0700 err: 0.000022 rank(L): 5 card(S): 767881
iter: 0710 err: 0.000021 rank(L): 5 card(S): 767926
iter: 0720 err: 0.000021 rank(L): 5 card(S): 767961
iter: 0730 err: 0.000020 rank(L): 5 card(S): 768003
iter: 0740 err: 0.000020 rank(L): 5 card(S): 768067
iter: 0750 err: 0.000019 rank(L): 5 card(S): 768115
iter: 0760 err: 0.000019 rank(L): 5 card(S): 768150
iter: 0770 err: 0.000018 rank(L): 5 card(S): 768203
iter: 0780 err: 0.000018 rank(L): 5 card(S): 768240
iter: 0790 err: 0.000018 rank(L): 5 card(S): 768276
iter: 0800 err: 0.000017 rank(L): 5 card(S): 768312
iter: 0810 err: 0.000017 rank(L): 5 card(S): 768338
iter: 0820 err: 0.000017 rank(L): 5 card(S): 768365
iter: 0830 err: 0.000017 rank(L): 5 card(S): 768393
iter: 0840 err: 0.000016 rank(L): 5 card(S): 768432
iter: 0850 err: 0.000016 rank(L): 5 card(S): 768466
iter: 0860 err: 0.000016 rank(L): 5 card(S): 768495
iter: 0870 err: 0.000015 rank(L): 5 card(S): 768537
iter: 0880 err: 0.000015 rank(L): 5 card(S): 768567
iter: 0890 err: 0.000015 rank(L): 5 card(S): 768603
iter: 0900 err: 0.000015 rank(L): 5 card(S): 768638
iter: 0910 err: 0.000014 rank(L): 5 card(S): 768660
iter: 0920 err: 0.000014 rank(L): 5 card(S): 768690
iter: 0930 err: 0.000014 rank(L): 5 card(S): 768714
iter: 0940 err: 0.000014 rank(L): 5 card(S): 768749
iter: 0950 err: 0.000013 rank(L): 5 card(S): 768776
iter: 0960 err: 0.000013 rank(L): 5 card(S): 768807
iter: 0970 err: 0.000013 rank(L): 5 card(S): 768826
iter: 0980 err: 0.000013 rank(L): 5 card(S): 768855
iter: 0990 err: 0.000012 rank(L): 5 card(S): 768880
iter: 1000 err: 0.000012 rank(L): 5 card(S): 768904
```



These very low-rank modes have some suggestive interpretations: * modes 1 and 2 describe discrete underlying states * mode 3 describes overall drift, which may be due to stress or simple chemical bleaching

```
plotSVD(L_filter_very_low(:,15:end),struct('sigma_modes',1:3));
title('3 svd modes (very low rank component)')
```



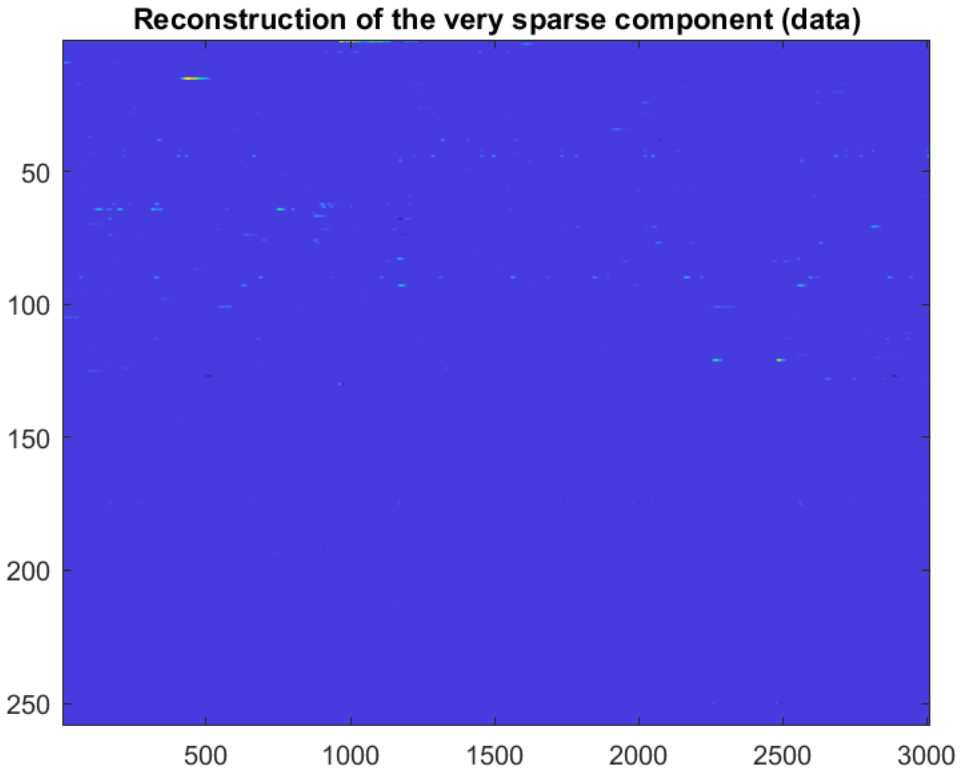
2nd RobustPCA, with a much more sparse matrix for S (higher value of lambda)

```
lambda = 0.05;
[L_high_rank, S_very_sparse] = RobustPCA(this_dat.', lambda);
```

```
iter: 0001 err: 0.098311 rank(L): 67 card(S): 4319
iter: 0010 err: 0.001151 rank(L): 258 card(S): 11672
iter: 0020 err: 0.000433 rank(L): 258 card(S): 11925
iter: 0030 err: 0.000166 rank(L): 258 card(S): 12041
iter: 0040 err: 0.000089 rank(L): 258 card(S): 12102
iter: 0050 err: 0.000067 rank(L): 258 card(S): 12114
iter: 0060 err: 0.000052 rank(L): 258 card(S): 12115
iter: 0070 err: 0.000038 rank(L): 258 card(S): 12125
iter: 0080 err: 0.000028 rank(L): 258 card(S): 12128
iter: 0090 err: 0.000019 rank(L): 258 card(S): 12132
iter: 0100 err: 0.000013 rank(L): 258 card(S): 12132
iter: 0110 err: 0.000009 rank(L): 258 card(S): 12135
iter: 0120 err: 0.000006 rank(L): 258 card(S): 12134
iter: 0130 err: 0.000004 rank(L): 258 card(S): 12132
iter: 0140 err: 0.000003 rank(L): 258 card(S): 12132
iter: 0150 err: 0.000002 rank(L): 258 card(S): 12132
iter: 0160 err: 0.000001 rank(L): 258 card(S): 12132
iter: 0164 err: 0.000001 rank(L): 258 card(S): 12132
```

Now plot the super sparse components that the algorithm has identified; some of the obvious sensory neurons are clearly picked out

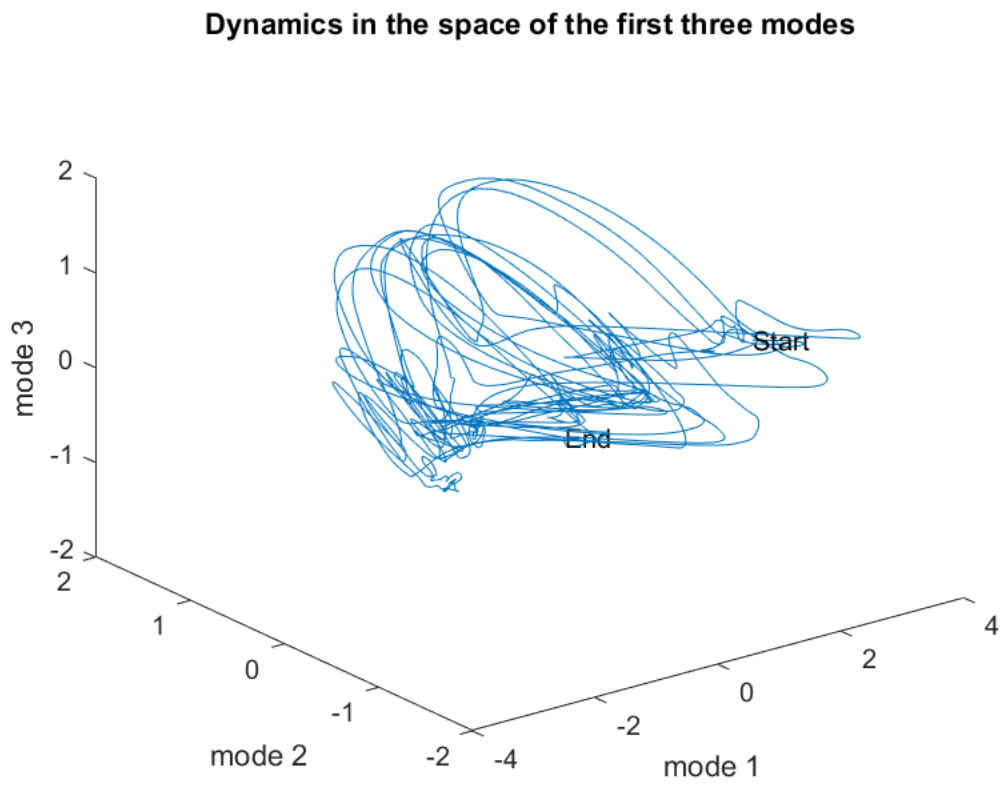
```
S_filter_very_sparse = my_filter(S_very_sparse, 10);
figure;
imagesc(S_filter_very_sparse.');
title('Reconstruction of the very sparse component (data)')
```

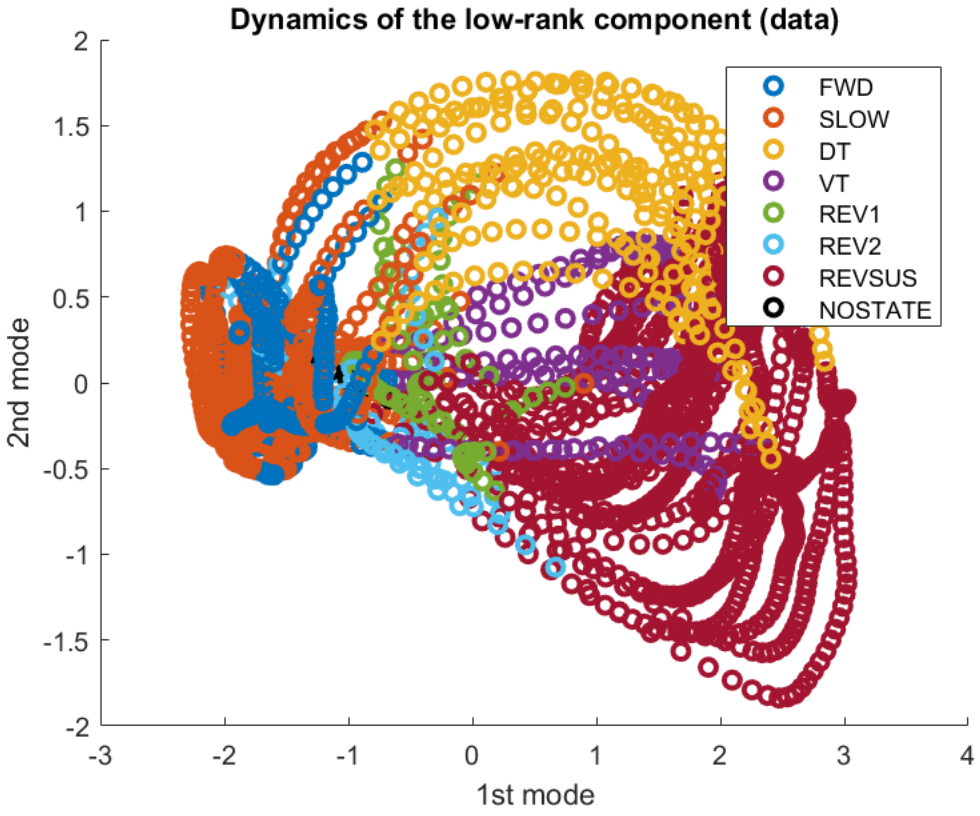


For a good 3d visualization, I've found that plotting the 'low-rank' component from the above algorithm (which takes out the sparse spikes) gives a very good 3d visualization, with separation between the different types of turns

This plots the regular 3d-pca projection as well as a more clear colored version

```
filter_window = 10;
L_filter_high_rank = my_filter(L_high_rank,filter_window)';
[~,~,~,proj3d] = plotSVD(L_filter_high_rank(:, filter_window:end),...
    struct('PCA3d',true, 'sigma',false));
plot_colored(proj3d,...
    dat_struct.SevenStates(2*filter_window-1:end),...
    dat_struct.SevenStatesKey,'o');
title('Dynamics of the low-rank component (data)')
```


Dynamics in the space of the first three modes
mode 3
mode 2
mode 1
Start
End


Dynamics of the low-rank component (data)
2nd mode
1st mode
FWD
SLOW
DT
VT
REV1
REV2
REVSUS
NOSTATE

For comparison, we can look at the 3d diagram produced by the reconstructed data. The first step is to get the AdaptiveDmdc object; see AdaptiveDmdc_documentation for a more thorough explanation:

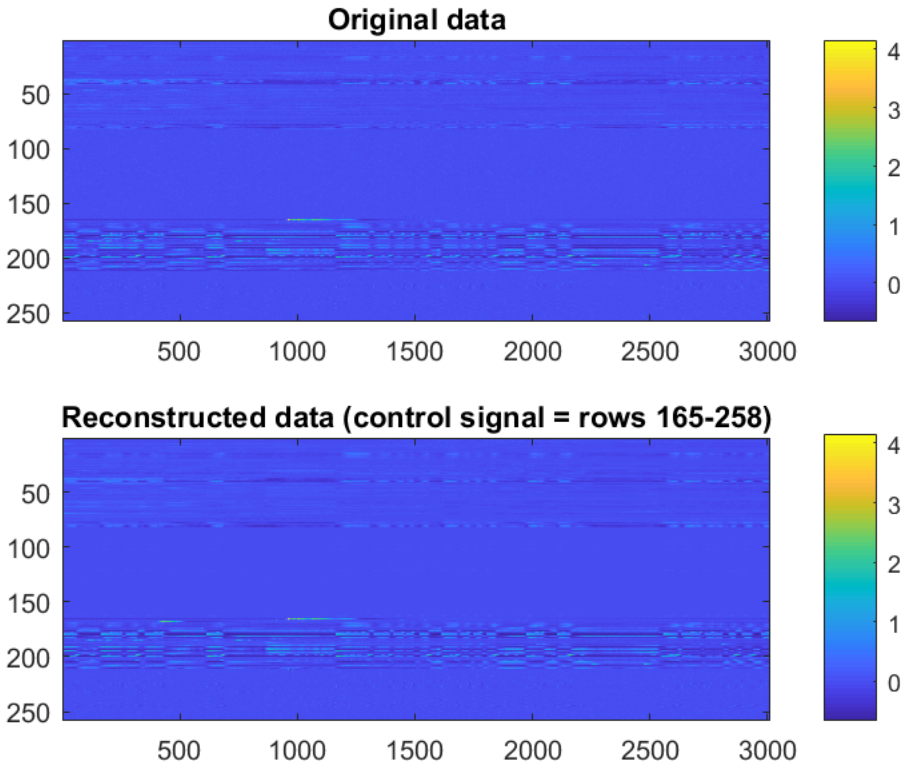
```
id_struct = struct(...
    'ID', {Zimmer_struct.ID},...
    'ID2', {Zimmer_struct.ID2},...
    'ID3', {Zimmer_struct.ID3});
settings = struct('to_normalize_envelope', true,...
    'to_subtract_mean',true,...
    'to_plot_nothing',true,...
    'id_struct',id_struct);

ad_obj2 = AdaptiveDmdc(this_dat, settings);
```

```
Preprocessing...
Finished analyzing
```

Use this object to reconstruct the data. First, plot it in comparison to the original data:

```
approx_data = ad_obj2.plot_reconstruction(true, true).';
```



Now use robust PCA and visualize this using the same algorithm as above

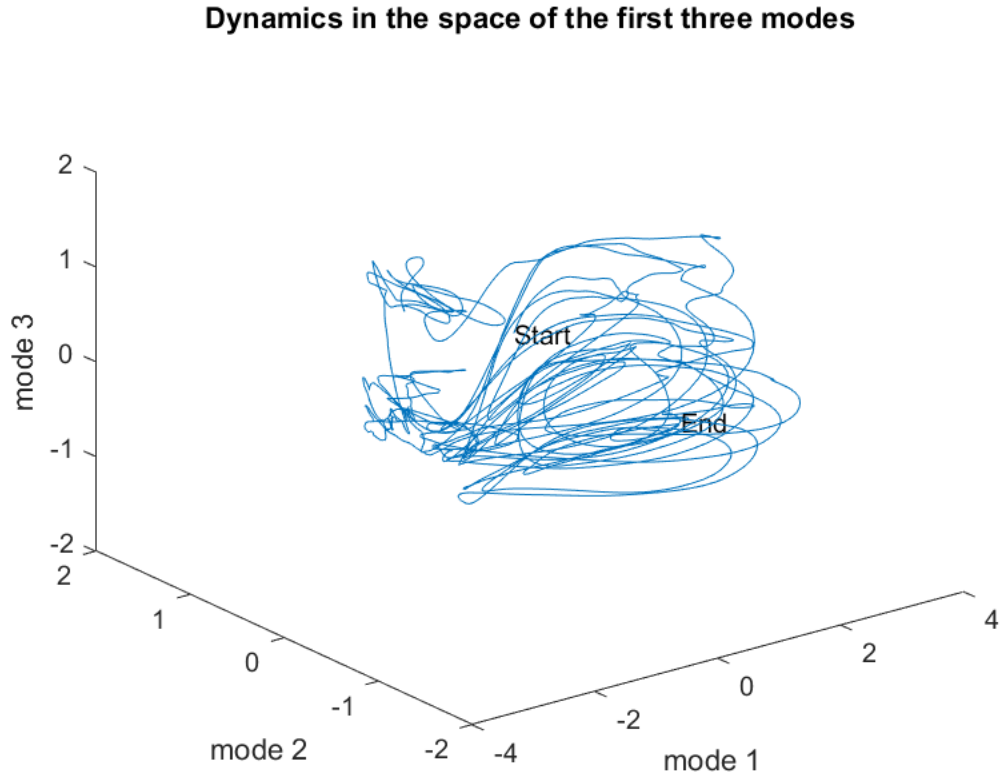
```
lambda = 0.05;
[L_reconstruct, S_reconstruct] = RobustPCA(approx_data, lambda);
% Plot the 2nd low-rank component
filter_window = 10;
L_filter2 = my_filter(L_reconstruct,filter_window)';
```

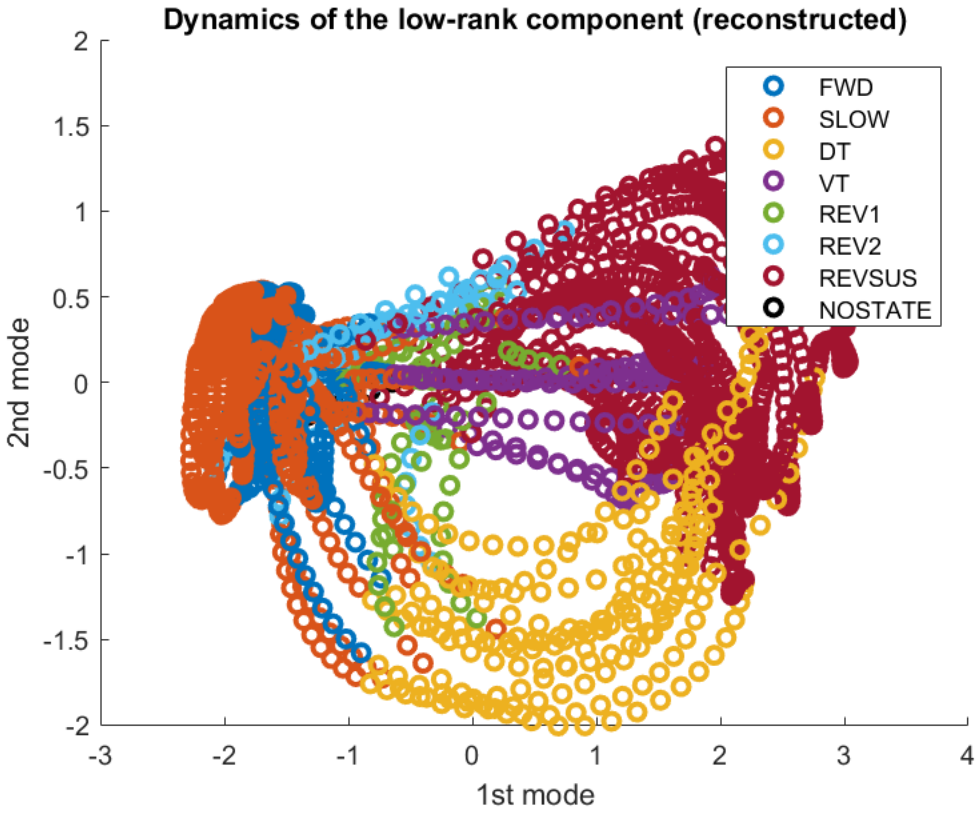
```
[u,s,v,proj3d] = plotSVD(L_filter2(:,filter_window:end),...
    struct('PCA3d',true,'sigma',false));
plot_colored(proj3d,...

 dat_struct.SevenStates(2*filter_window-1:end),dat_struct.SevenStatesKey,'o');
title('Dynamics of the low-rank component (reconstructed)')
%=========================================================================
```

```
iter: 0001 err: 0.124396 rank(L): 52 card(S): 3313
iter: 0010 err: 0.007845 rank(L): 142 card(S): 7965
iter: 0020 err: 0.003731 rank(L): 169 card(S): 9262
iter: 0030 err: 0.002304 rank(L): 187 card(S): 9995
iter: 0040 err: 0.001675 rank(L): 198 card(S): 10535
iter: 0050 err: 0.001262 rank(L): 207 card(S): 10997
iter: 0060 err: 0.000998 rank(L): 214 card(S): 11378
iter: 0070 err: 0.000838 rank(L): 219 card(S): 11671
iter: 0080 err: 0.000717 rank(L): 223 card(S): 11923
iter: 0090 err: 0.000589 rank(L): 228 card(S): 12150
iter: 0100 err: 0.000536 rank(L): 230 card(S): 12380
iter: 0110 err: 0.000473 rank(L): 233 card(S): 12524
iter: 0120 err: 0.000415 rank(L): 236 card(S): 12665
iter: 0130 err: 0.000354 rank(L): 240 card(S): 12782
iter: 0140 err: 0.000320 rank(L): 241 card(S): 12892
iter: 0150 err: 0.000295 rank(L): 243 card(S): 12991
iter: 0160 err: 0.000265 rank(L): 244 card(S): 13067
iter: 0170 err: 0.000232 rank(L): 246 card(S): 13140
iter: 0180 err: 0.000199 rank(L): 248 card(S): 13221
iter: 0190 err: 0.000183 rank(L): 249 card(S): 13284
iter: 0200 err: 0.000165 rank(L): 250 card(S): 13339
iter: 0210 err: 0.000164 rank(L): 250 card(S): 13377
iter: 0220 err: 0.000148 rank(L): 251 card(S): 13419
iter: 0230 err: 0.000122 rank(L): 253 card(S): 13462
iter: 0240 err: 0.000116 rank(L): 253 card(S): 13497
iter: 0250 err: 0.000115 rank(L): 253 card(S): 13531
iter: 0260 err: 0.000100 rank(L): 254 card(S): 13547
iter: 0270 err: 0.000084 rank(L): 255 card(S): 13563
iter: 0280 err: 0.000074 rank(L): 256 card(S): 13586
iter: 0290 err: 0.000065 rank(L): 256 card(S): 13597
iter: 0300 err: 0.000064 rank(L): 256 card(S): 13613
iter: 0310 err: 0.000064 rank(L): 256 card(S): 13619
iter: 0320 err: 0.000048 rank(L): 257 card(S): 13632
iter: 0330 err: 0.000044 rank(L): 257 card(S): 13639
iter: 0340 err: 0.000043 rank(L): 257 card(S): 13652
iter: 0350 err: 0.000012 rank(L): 258 card(S): 13666
iter: 0360 err: 0.000011 rank(L): 258 card(S): 13678
iter: 0370 err: 0.000010 rank(L): 258 card(S): 13686
iter: 0380 err: 0.000009 rank(L): 258 card(S): 13698
iter: 0390 err: 0.000008 rank(L): 258 card(S): 13702
iter: 0400 err: 0.000008 rank(L): 258 card(S): 13709
iter: 0410 err: 0.000007 rank(L): 258 card(S): 13718
iter: 0420 err: 0.000007 rank(L): 258 card(S): 13720
iter: 0430 err: 0.000006 rank(L): 258 card(S): 13719
iter: 0440 err: 0.000006 rank(L): 258 card(S): 13718
iter: 0450 err: 0.000006 rank(L): 258 card(S): 13720
```

```
iter: 0460 err: 0.000005 rank(L): 258 card(S): 13722
iter: 0470 err: 0.000005 rank(L): 258 card(S): 13720
iter: 0480 err: 0.000005 rank(L): 258 card(S): 13721
iter: 0490 err: 0.000004 rank(L): 258 card(S): 13723
iter: 0500 err: 0.000004 rank(L): 258 card(S): 13724
iter: 0510 err: 0.000004 rank(L): 258 card(S): 13725
iter: 0520 err: 0.000004 rank(L): 258 card(S): 13725
iter: 0530 err: 0.000003 rank(L): 258 card(S): 13726
iter: 0540 err: 0.000003 rank(L): 258 card(S): 13727
iter: 0550 err: 0.000003 rank(L): 258 card(S): 13727
iter: 0560 err: 0.000003 rank(L): 258 card(S): 13727
iter: 0570 err: 0.000003 rank(L): 258 card(S): 13728
iter: 0580 err: 0.000003 rank(L): 258 card(S): 13730
iter: 0590 err: 0.000003 rank(L): 258 card(S): 13730
iter: 0600 err: 0.000002 rank(L): 258 card(S): 13730
iter: 0610 err: 0.000002 rank(L): 258 card(S): 13729
iter: 0620 err: 0.000002 rank(L): 258 card(S): 13731
iter: 0630 err: 0.000002 rank(L): 258 card(S): 13731
iter: 0640 err: 0.000002 rank(L): 258 card(S): 13732
iter: 0650 err: 0.000002 rank(L): 258 card(S): 13732
iter: 0660 err: 0.000002 rank(L): 258 card(S): 13733
iter: 0670 err: 0.000002 rank(L): 258 card(S): 13733
iter: 0680 err: 0.000002 rank(L): 258 card(S): 13734
iter: 0690 err: 0.000002 rank(L): 258 card(S): 13736
iter: 0700 err: 0.000001 rank(L): 258 card(S): 13736
iter: 0710 err: 0.000001 rank(L): 258 card(S): 13737
iter: 0720 err: 0.000001 rank(L): 258 card(S): 13737
iter: 0730 err: 0.000001 rank(L): 258 card(S): 13738
iter: 0740 err: 0.000001 rank(L): 258 card(S): 13736
iter: 0750 err: 0.000001 rank(L): 258 card(S): 13736
iter: 0760 err: 0.000001 rank(L): 258 card(S): 13736
iter: 0770 err: 0.000001 rank(L): 258 card(S): 13736
iter: 0780 err: 0.000001 rank(L): 258 card(S): 13738
iter: 0790 err: 0.000001 rank(L): 258 card(S): 13738
iter: 0791 err: 0.000001 rank(L): 258 card(S): 13738
```


Dynamics in the space of the first three modes
mode 3
mode 2
mode 1
Start
End


Dynamics of the low-rank component (reconstructed)
2nd mode
1st mode
FWD
SLOW
DT
VT
REV1
REV2
REVSUS
NOSTATE

Published with MATLAB® R2017a