

phoneme_analysis

July 11, 2017

```
In [1]: import timit_stats as ts
gestures_scaled = ts.load_obj("scaled_gestures")

In [3]: import gesture as ges
import os
import matplotlib.pyplot as plt

%pylab inline

articulators = ["LL", "UL", "TT", "TB", "TD", "JAW"]
domains = ["_x", "_y"]

for g in gestures_scaled:
    phonemes = gestures_scaled[g].phonemes
    print g
    for d in domains:
        fig1, ax = plt.subplots(1, len(articulators), figsize=(20, 4))

        fig1.suptitle("Phoneme \"{}\" in \"{}\" domain".format(g, d),
                      fontsize=10, fontweight='bold')
        for i in range(len(articulators)):
            ax[i].grid(color='black', linestyle='-', linewidth=1, alpha=0.3, axis='xy')
            ax[i].set_xlabel("time, (sec)")
            ax[i].set_ylabel("position, (mm)")
            ax[i].legend(loc='lower left')
            ax[i].title.set_text(articulators[i]+d)
            t = np.linspace(0, 100, 100)
            mean = [0.0]*100
            variance = [0.0]*100
            skipped = 0
            for phoneme in phonemes:
                ax[i].plot(t, phoneme.params[articulators[i]+d], color="black", linewidth=1)
                if any([math.isnan(v) for v in phoneme.params[articulators[i]+d]]):
                    skipped += 1
                    continue
                mean = [mean[j] + phoneme.params[articulators[i]+d][j] for j in range(len(phoneme.params[articulators[i]+d]))]
```

```

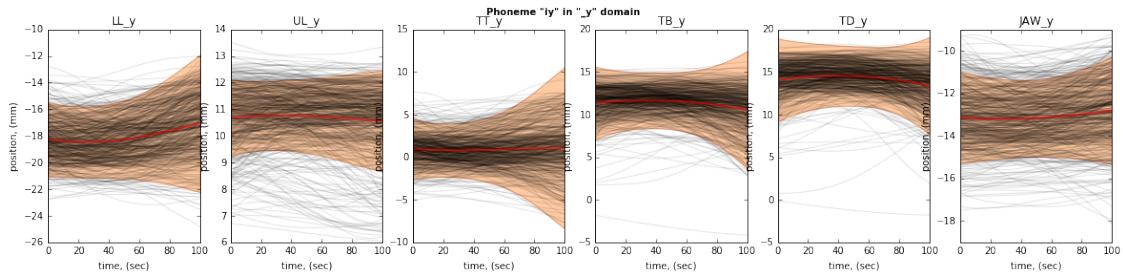
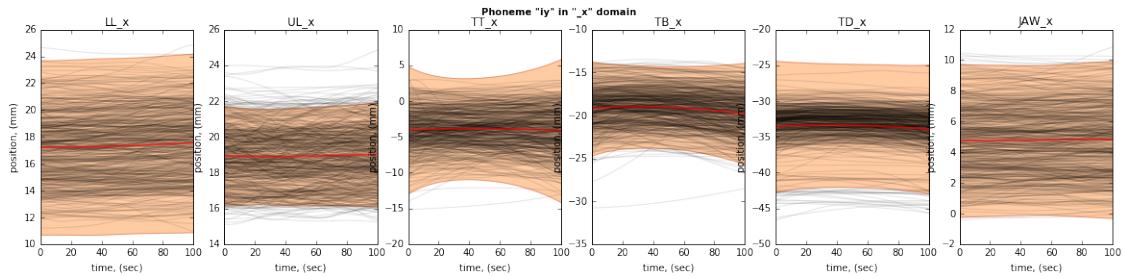
variance = [variance[j] + phoneme.params[articulators[i]+d][j]**2 for
            j in range(len(phoneme.params[articulators[i]+d]))]
mean = [mean[j]/(len(phonemes) - skipped) for j in range(len(mean))]
variance = [variance[j]/(len(phonemes) - skipped) for j in range(len(variance))]
variance = [(variance[j] - mean[j]**2)*1.0 for j in range(len(variance))]
mean_upper = [mean[j]+variance[j] for j in range(len(mean))]
mean_lower = [mean[j]-variance[j] for j in range(len(mean))]

ax[i].plot(t, mean, color="red", linewidth=2, alpha=0.6)
ax[i].fill_between(t, mean_lower, mean_upper,
                    alpha=0.5, edgecolor='#CC4F1B', facecolor='#FF9848')
plt.show()

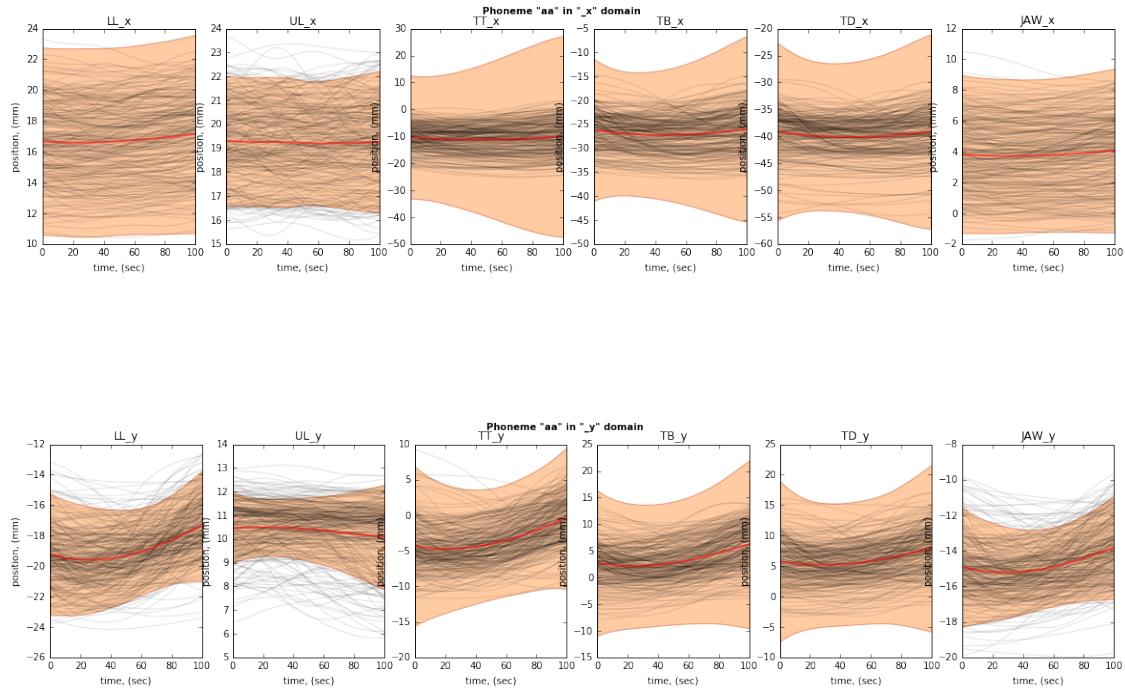
```

Populating the interactive namespace from numpy and matplotlib
iy

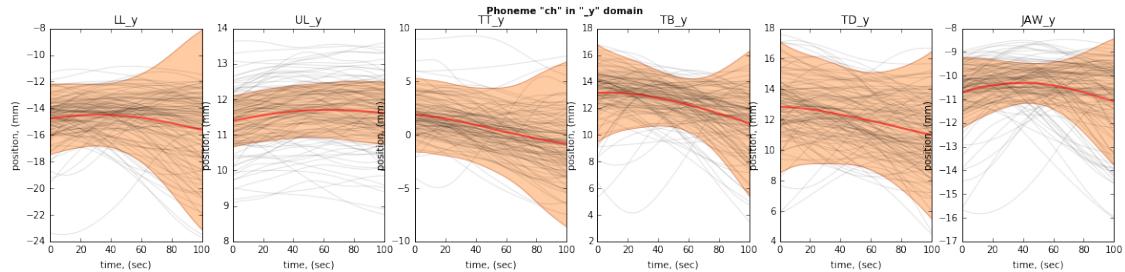
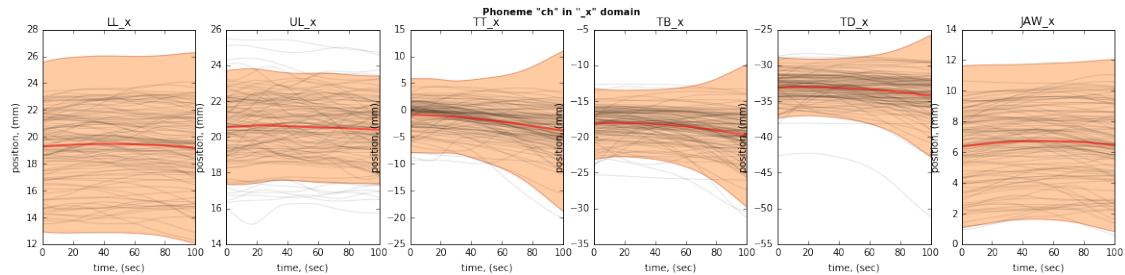
WARNING: pylab import has clobbered these variables: ['mean']
`%matplotlib` prevents importing * from pylab and numpy



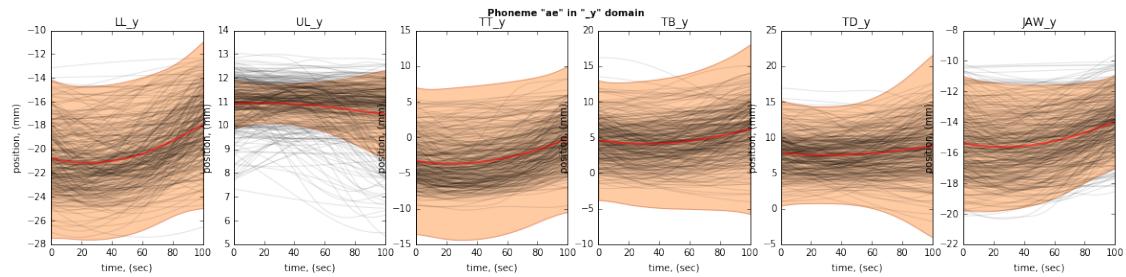
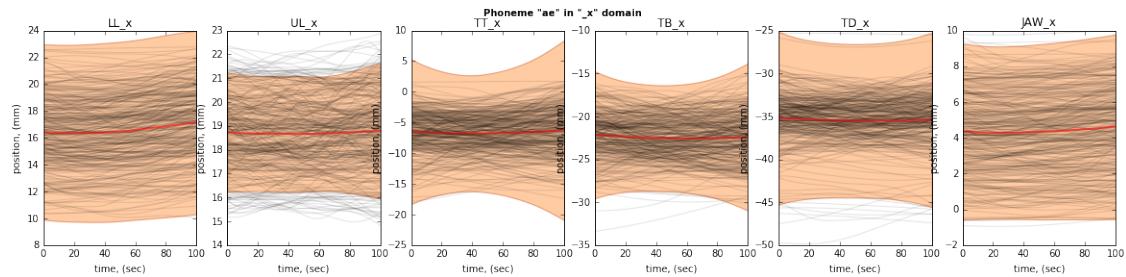
aa



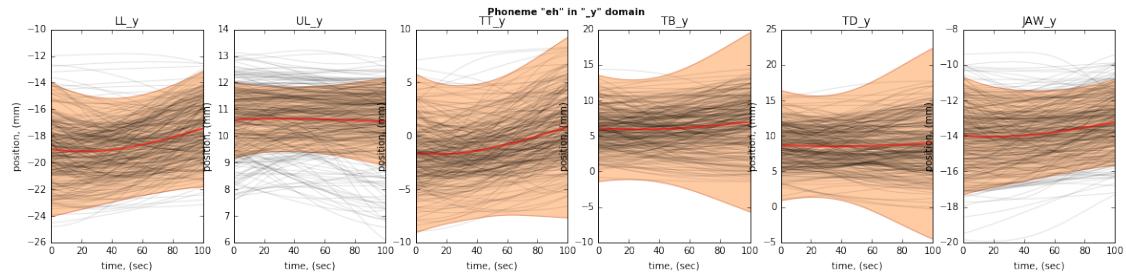
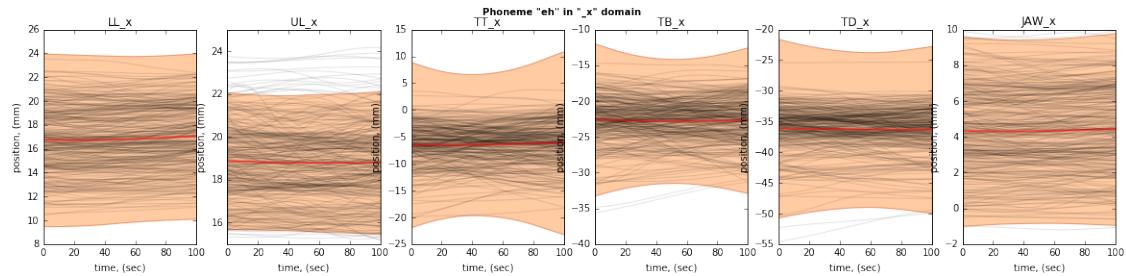
ch



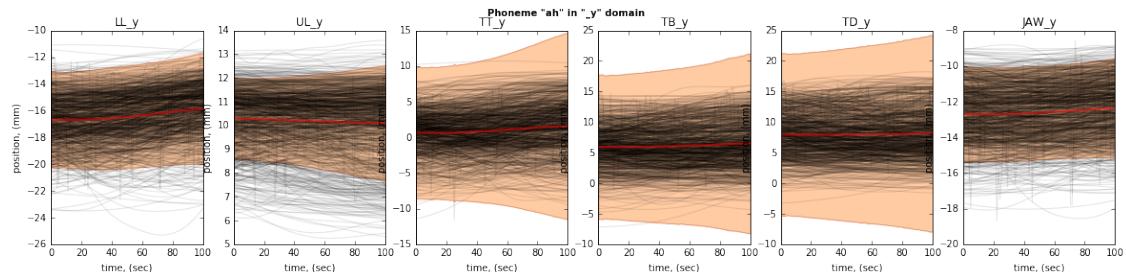
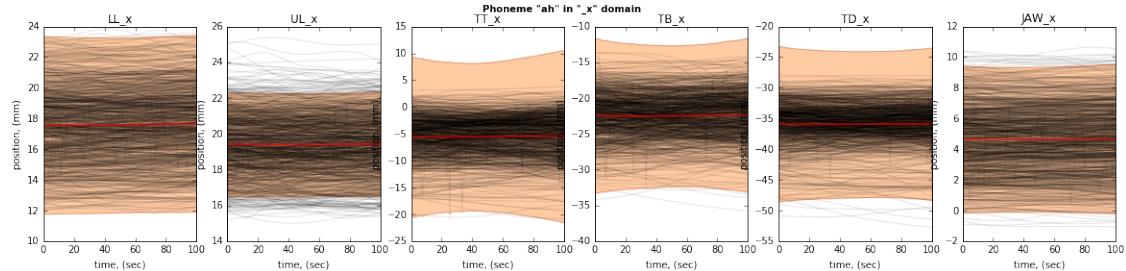
ae



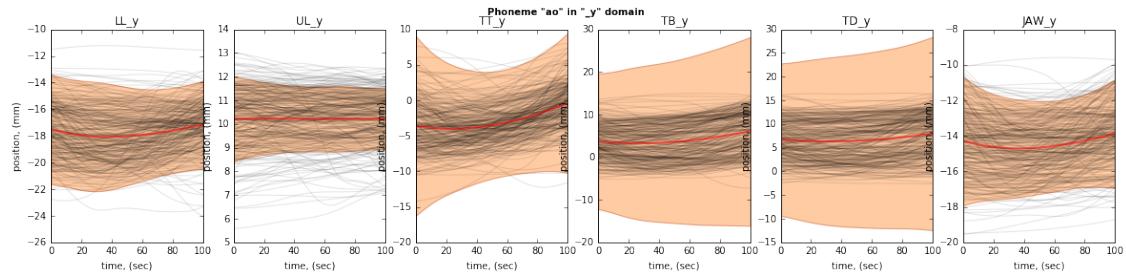
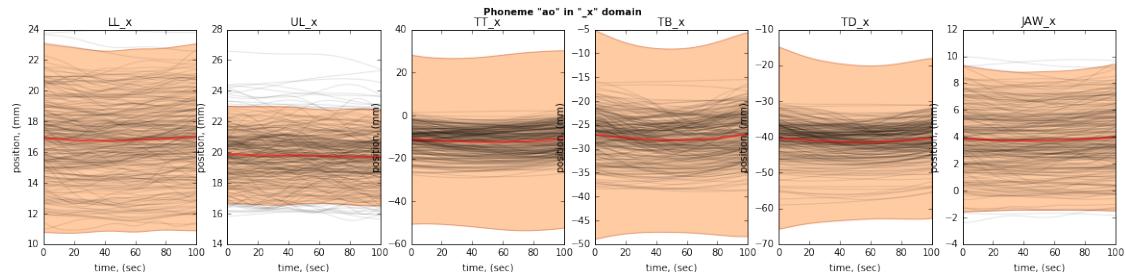
eh



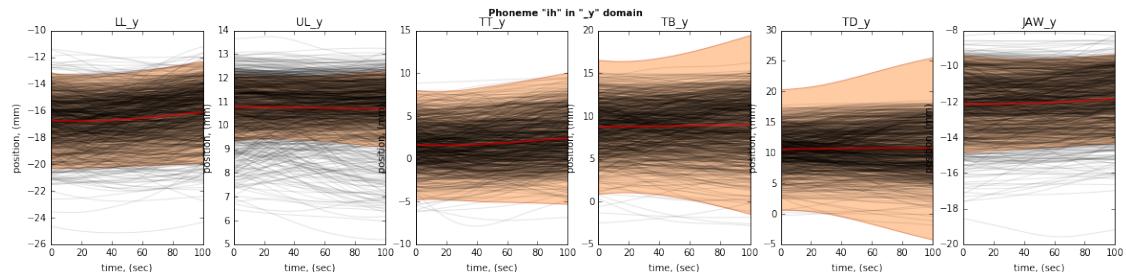
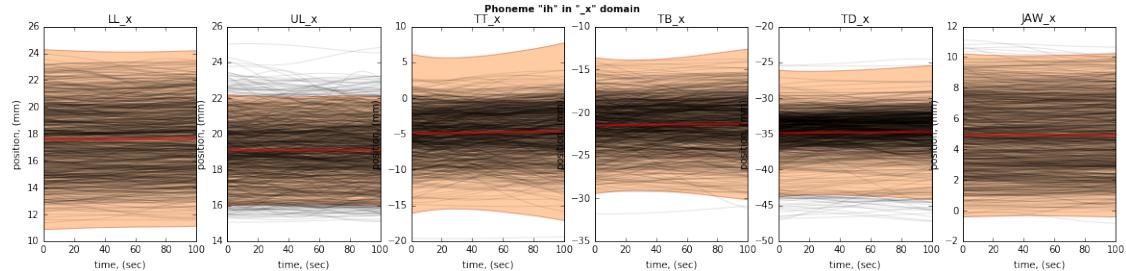
ah



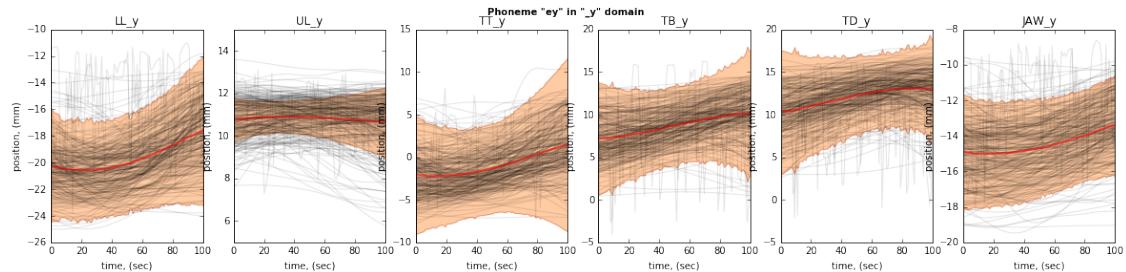
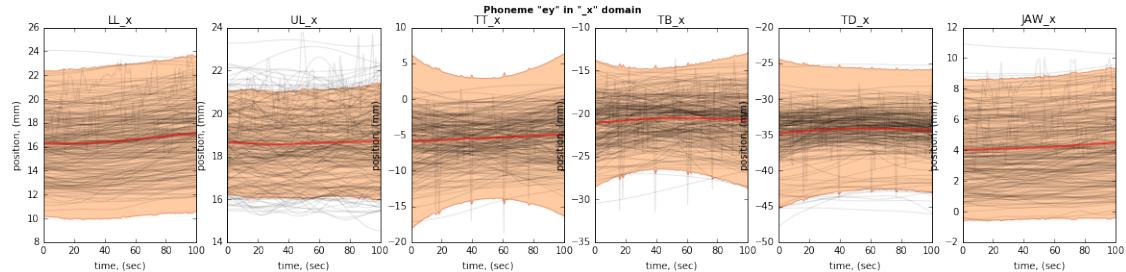
ao



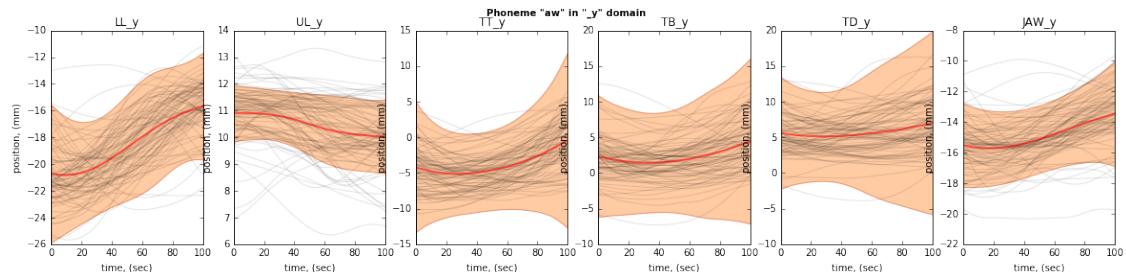
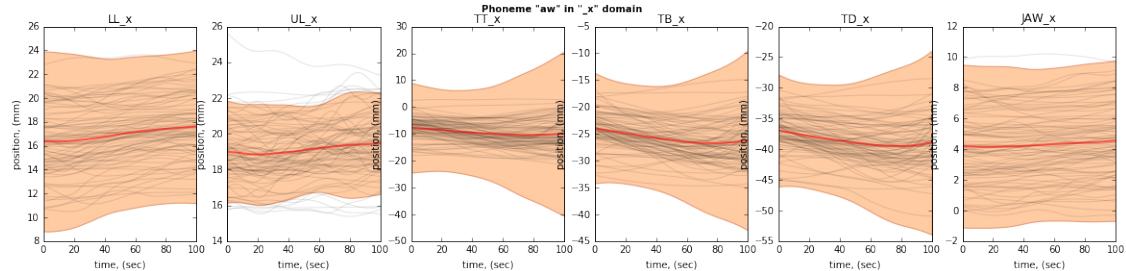
ih



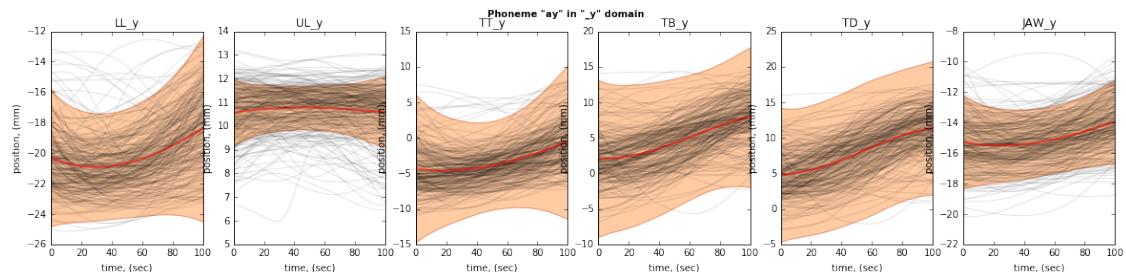
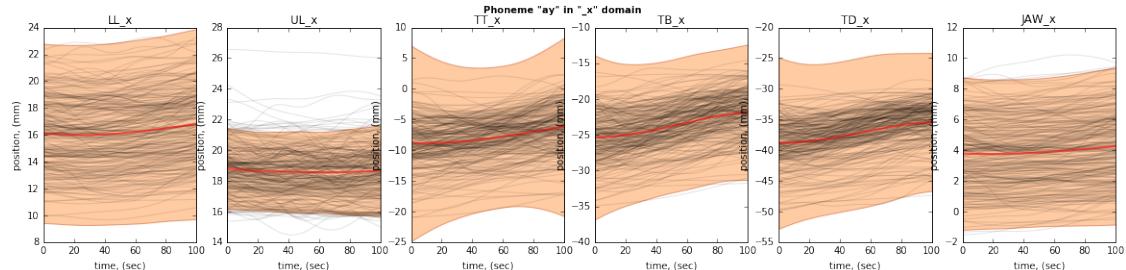
ey



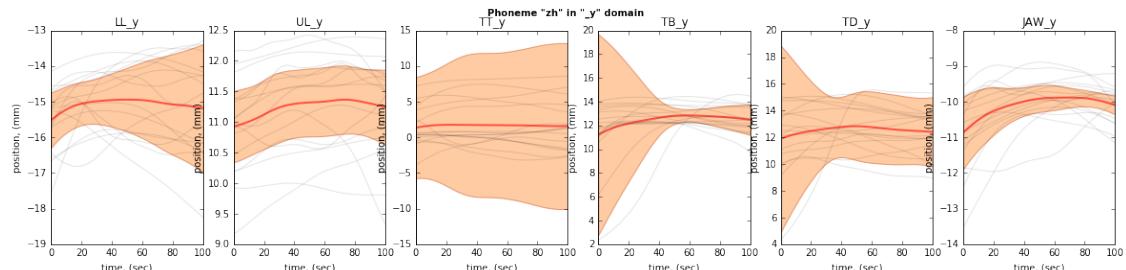
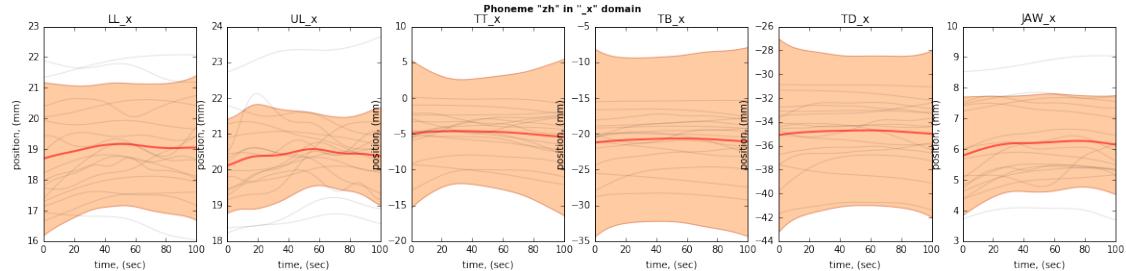
aw



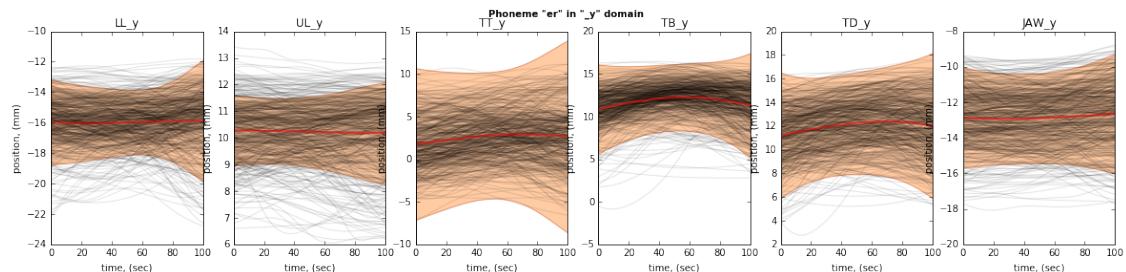
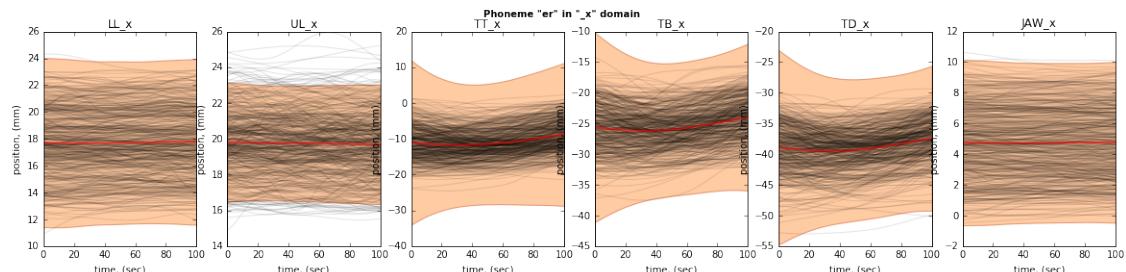
ay



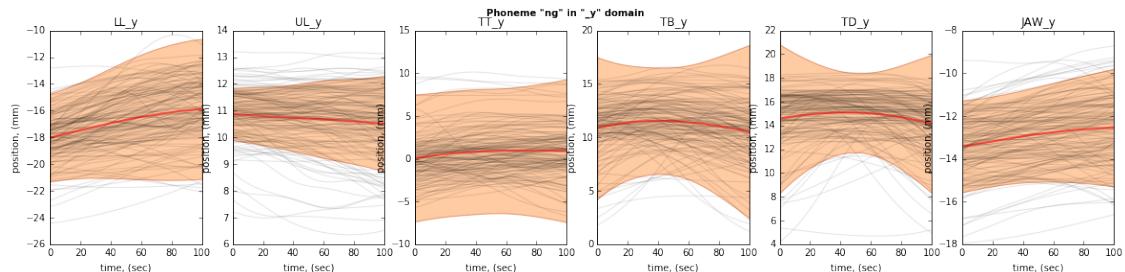
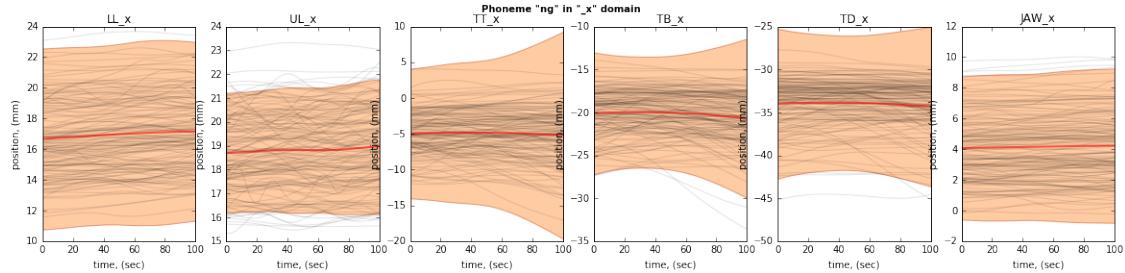
zh



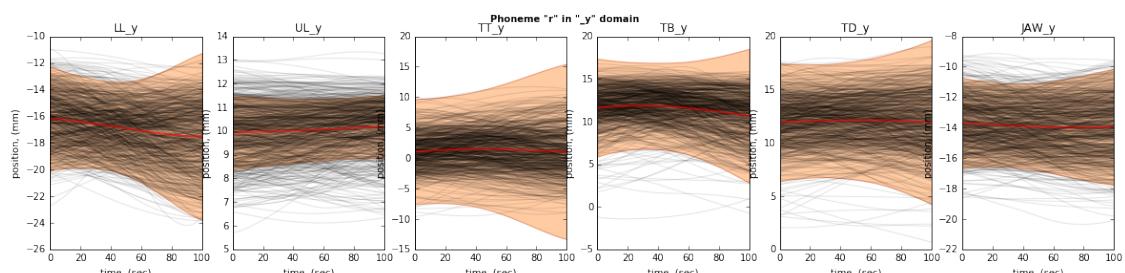
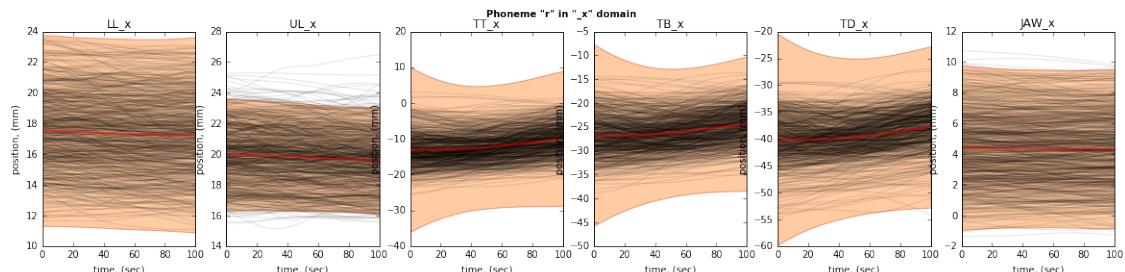
er



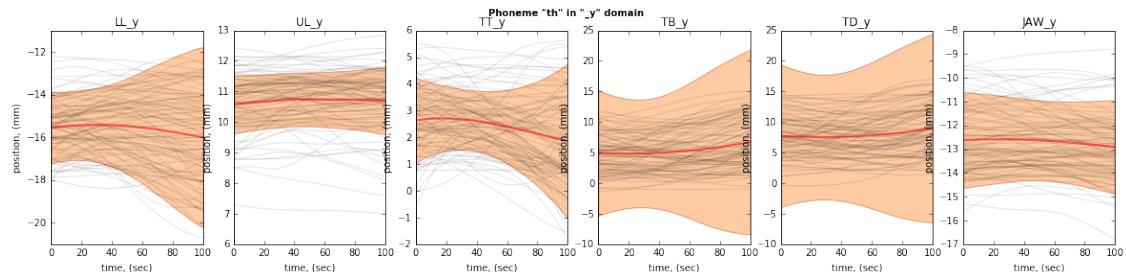
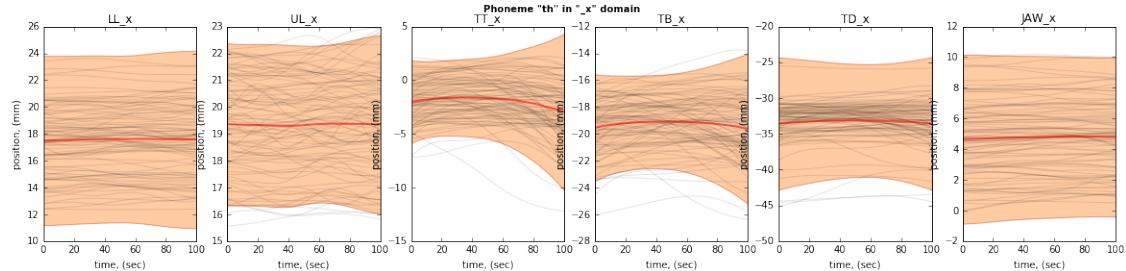
ng



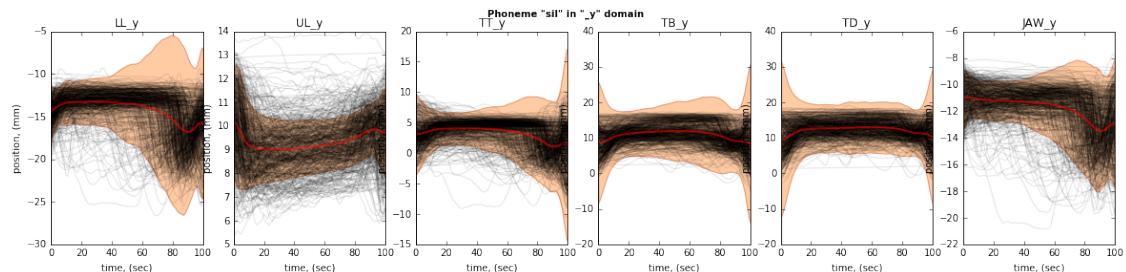
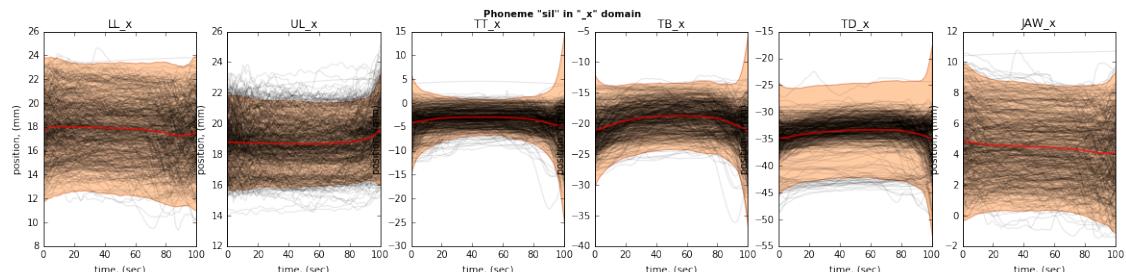
r



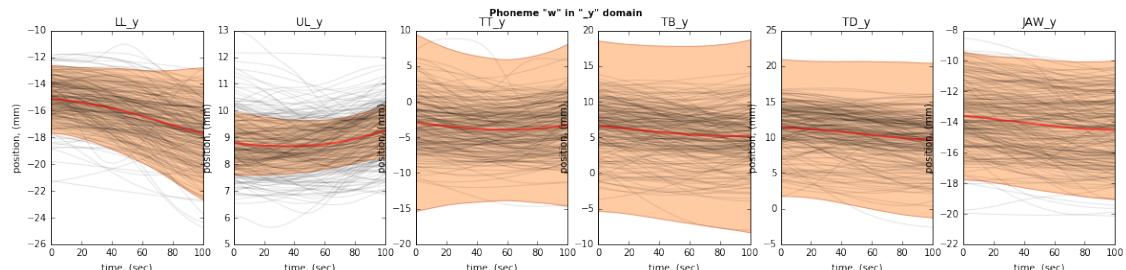
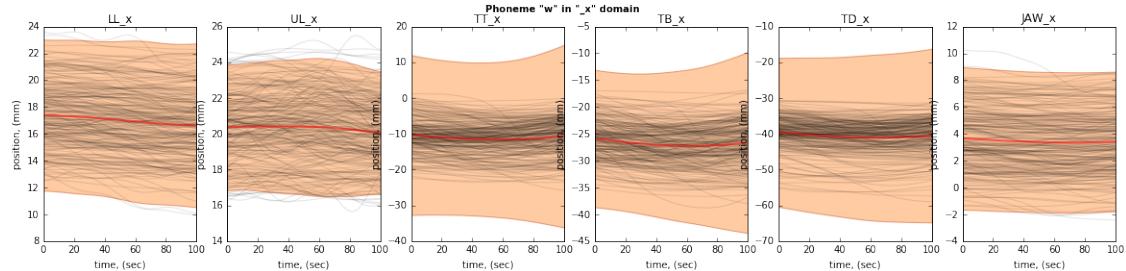
th



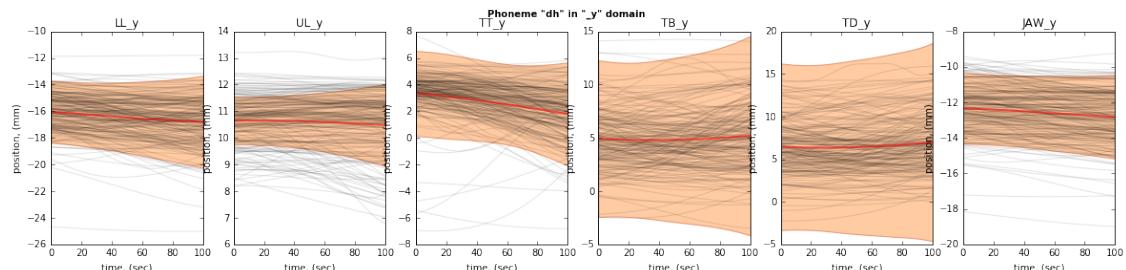
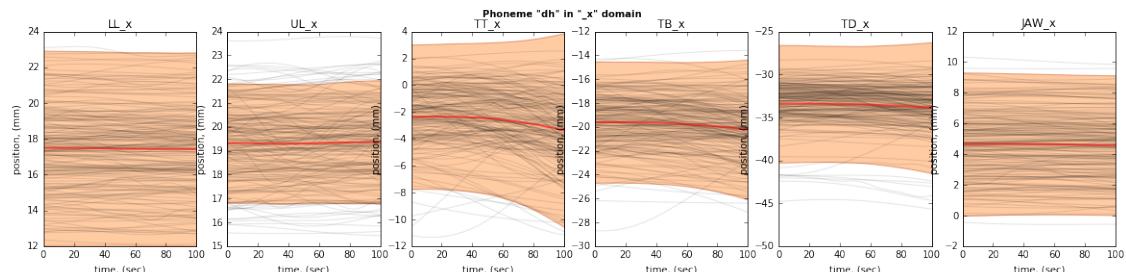
sil



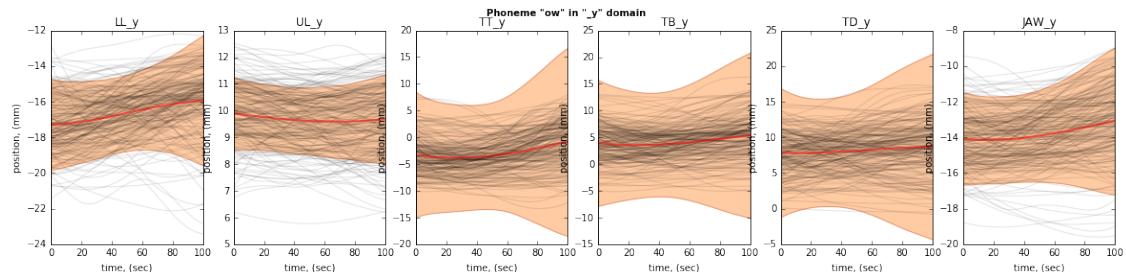
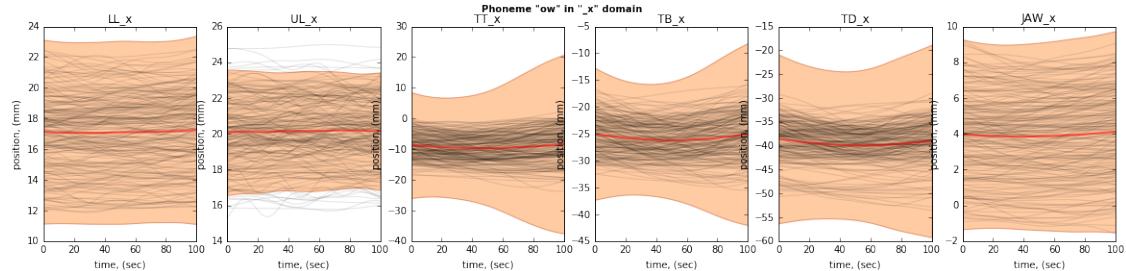
W



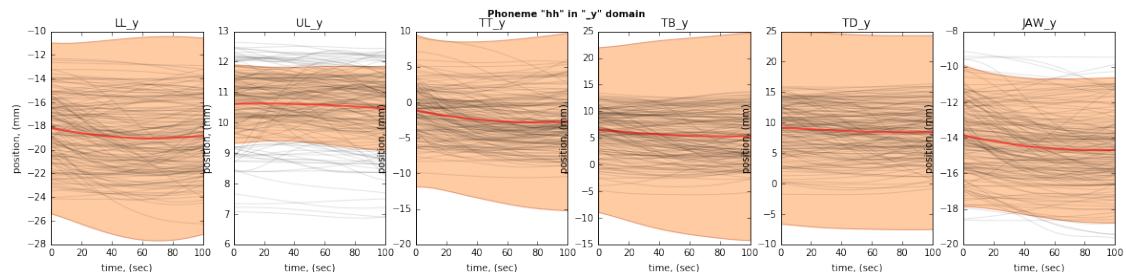
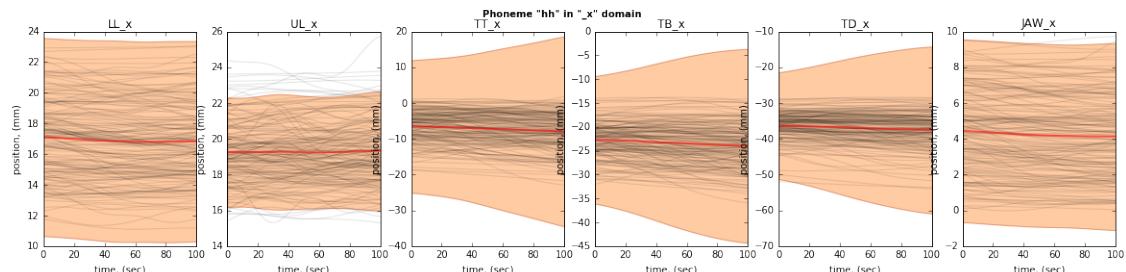
dh



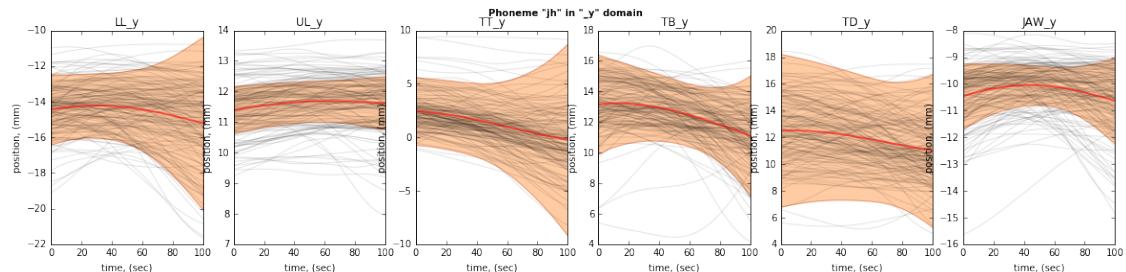
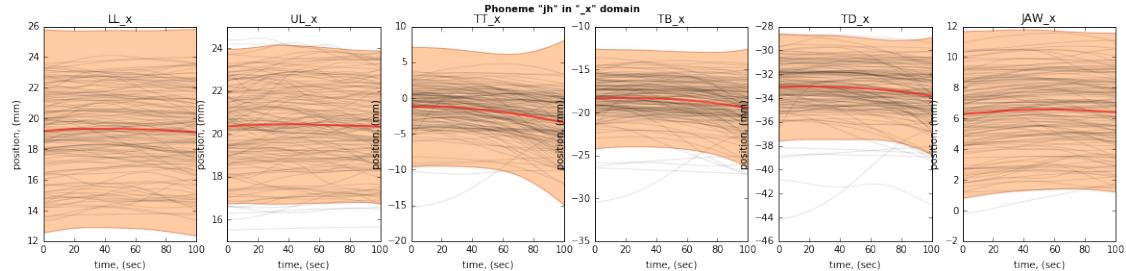
ow



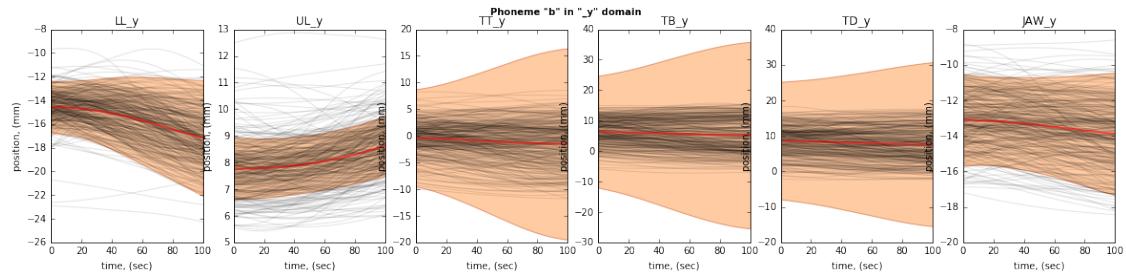
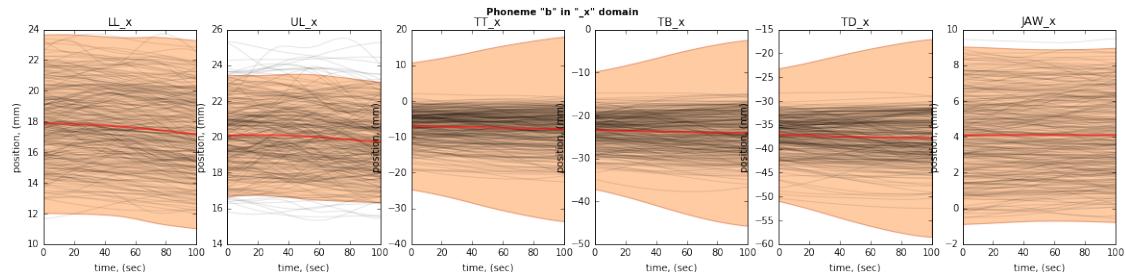
hh



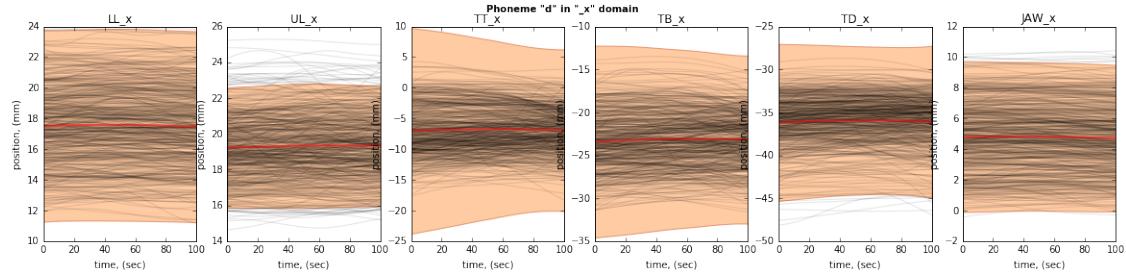
jh



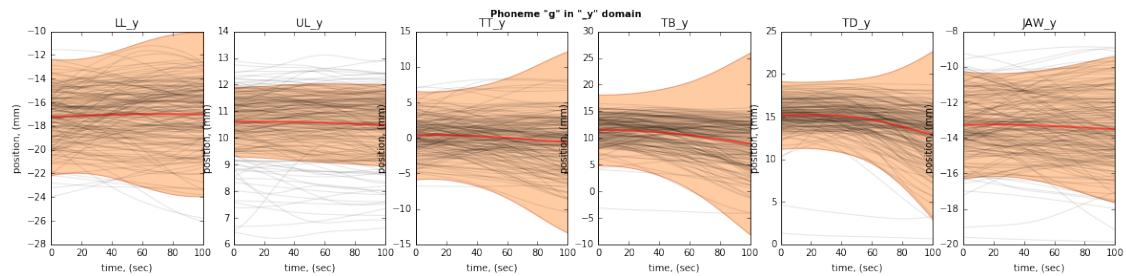
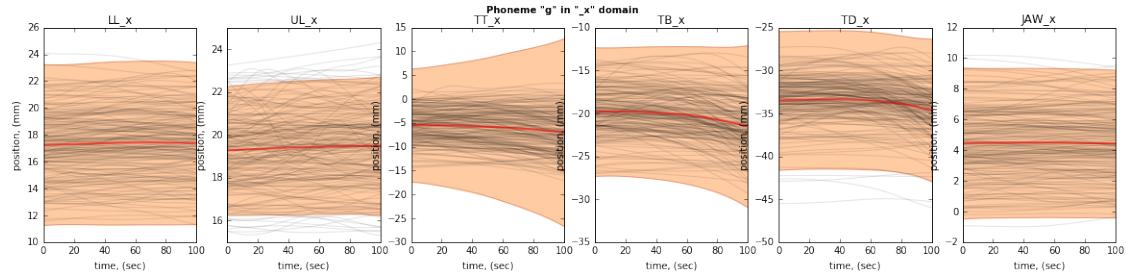
b



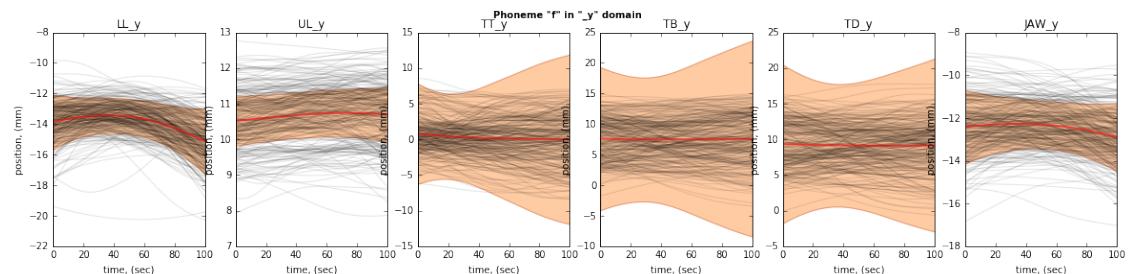
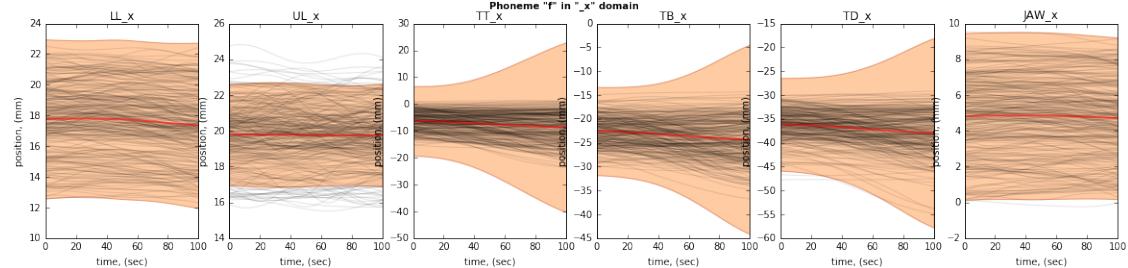
d



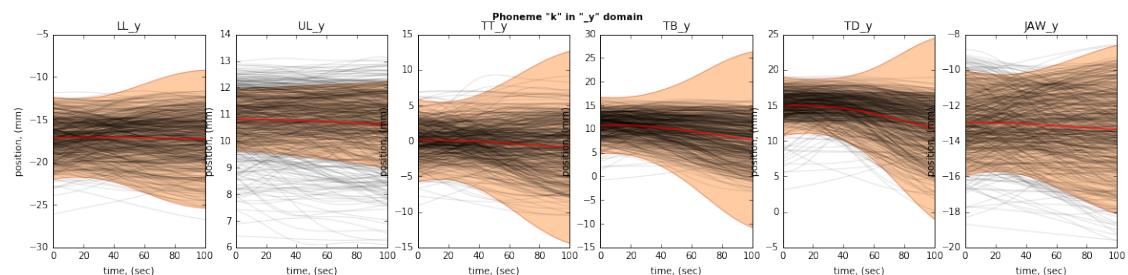
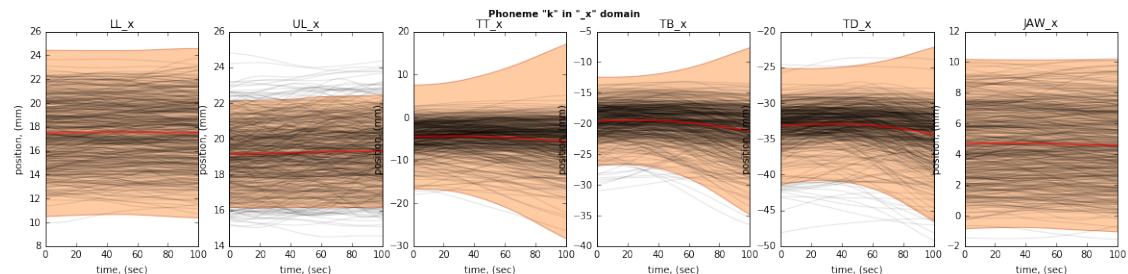
g



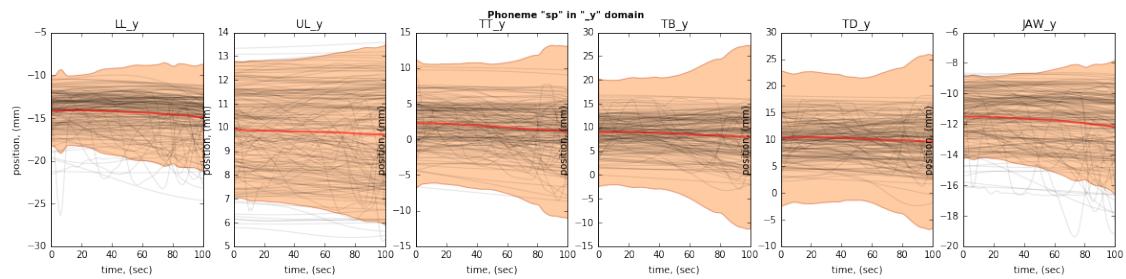
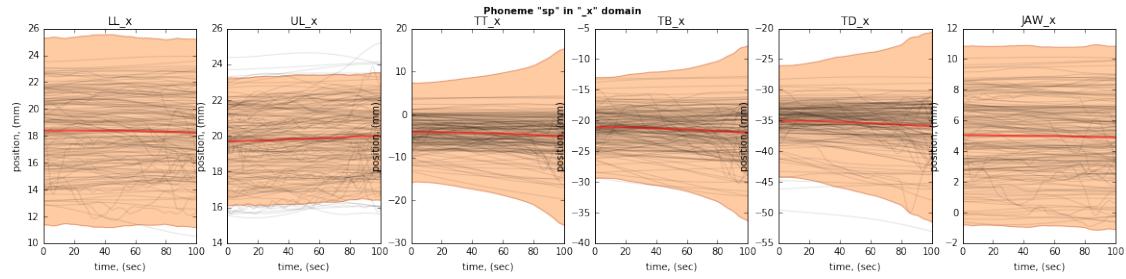
f



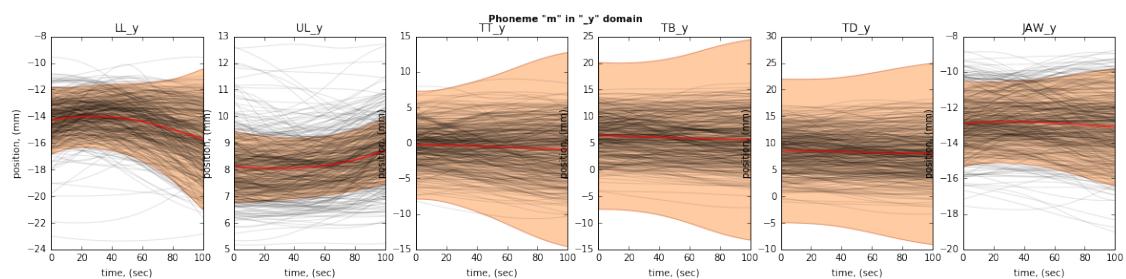
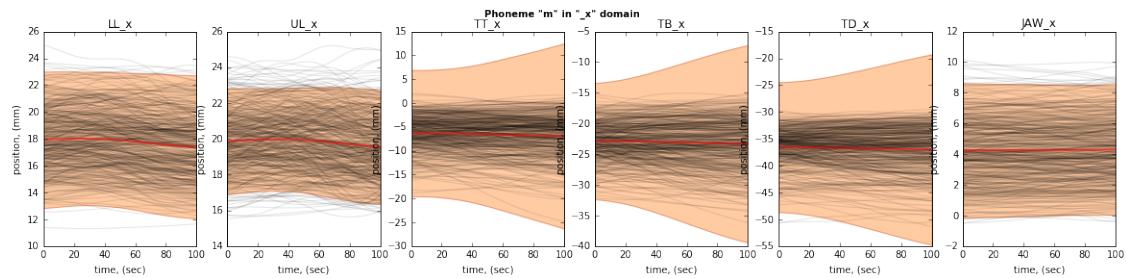
k



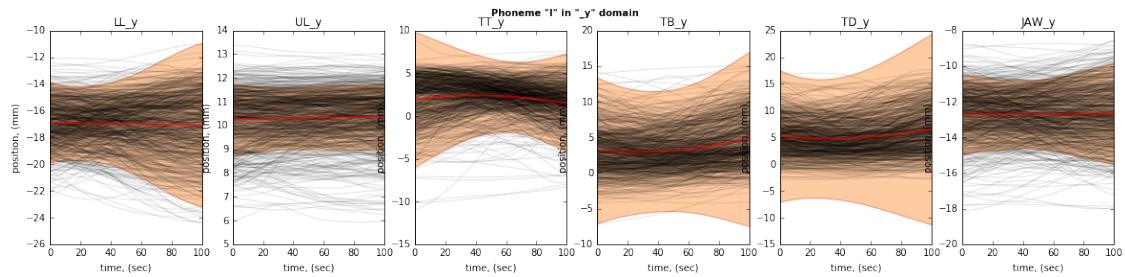
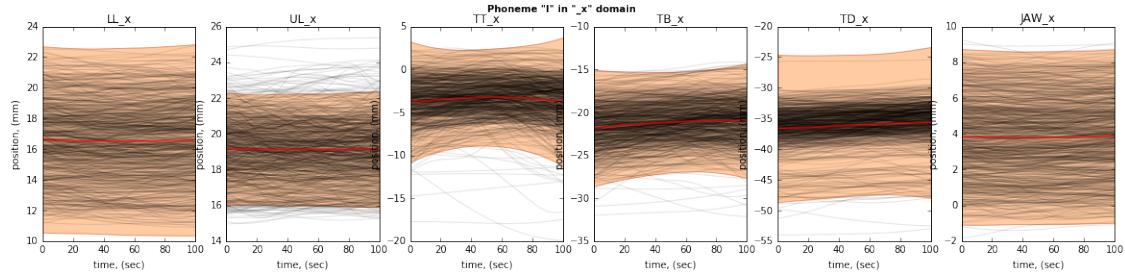
sp



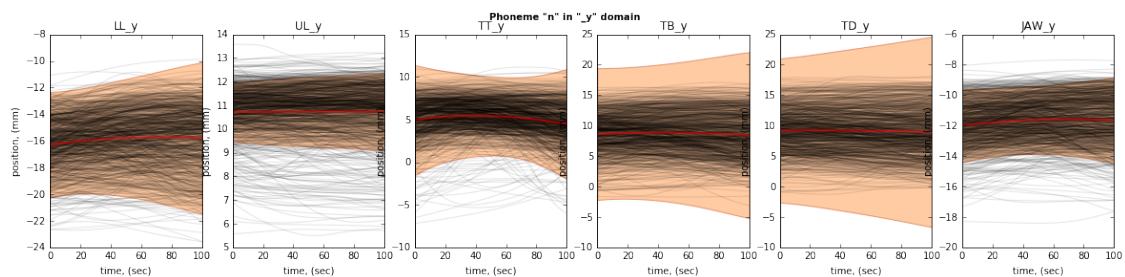
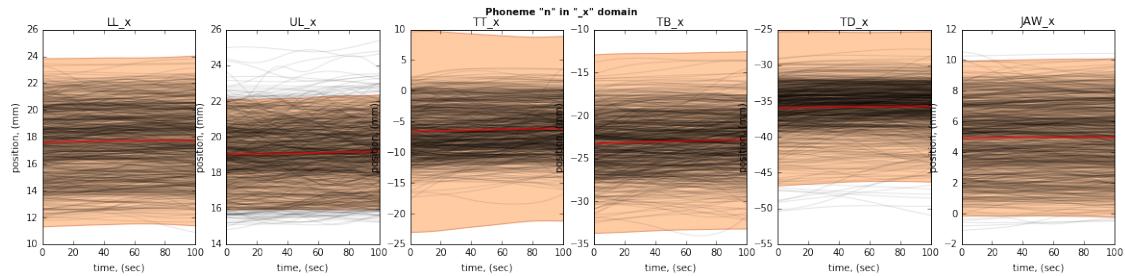
m



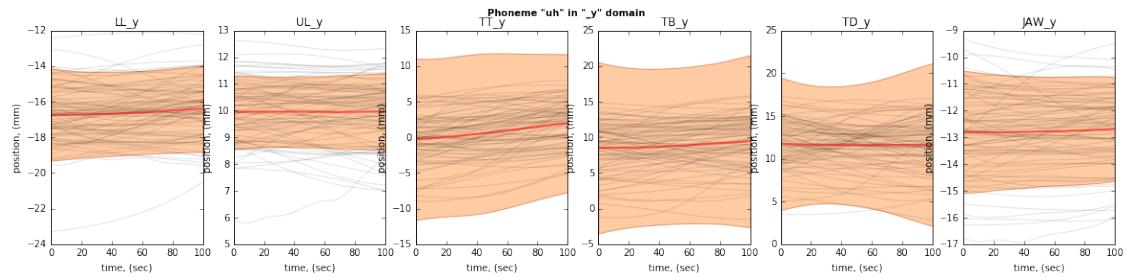
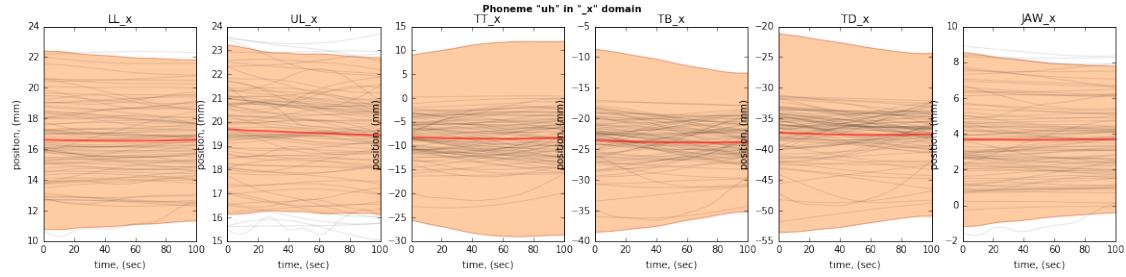
1



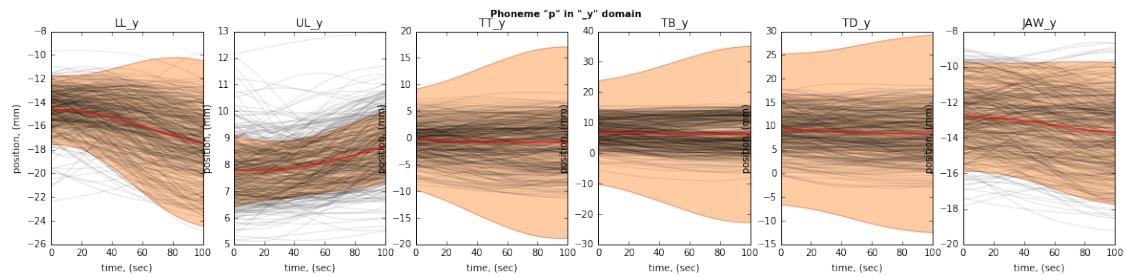
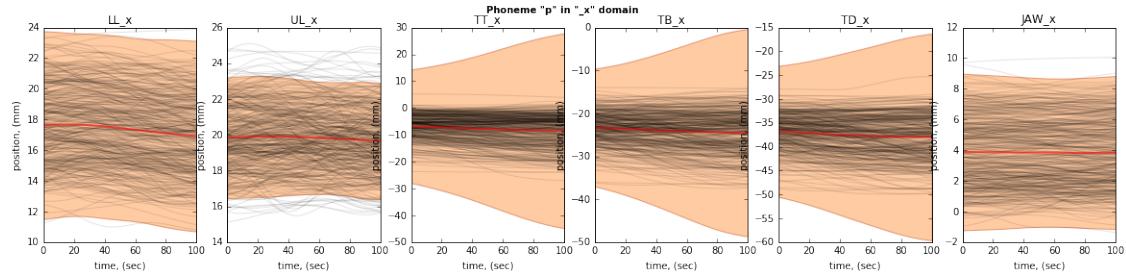
n



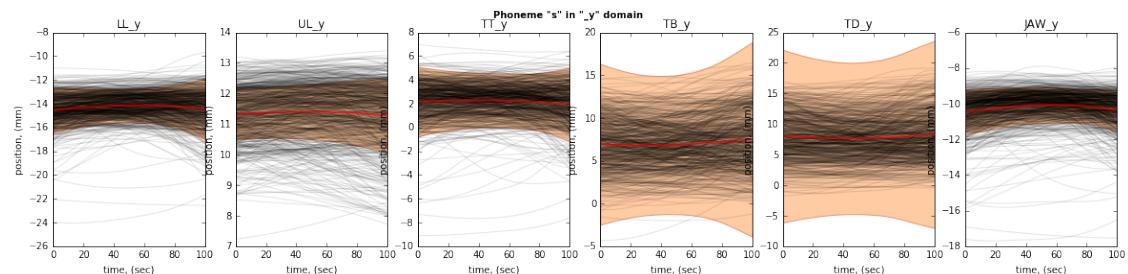
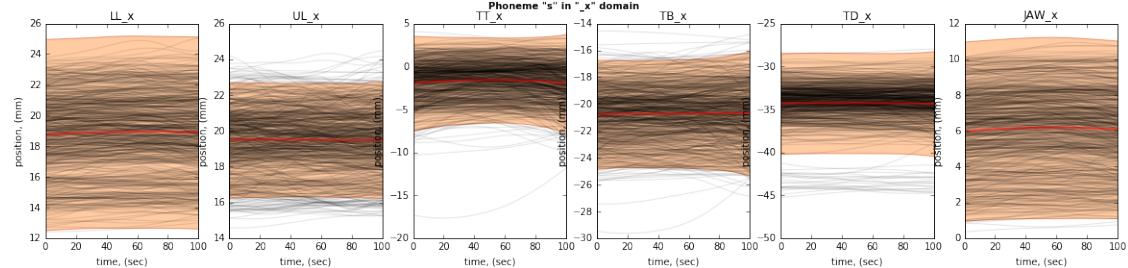
uh



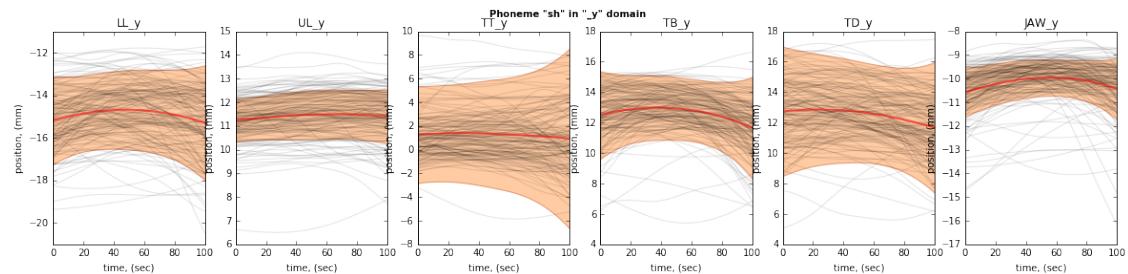
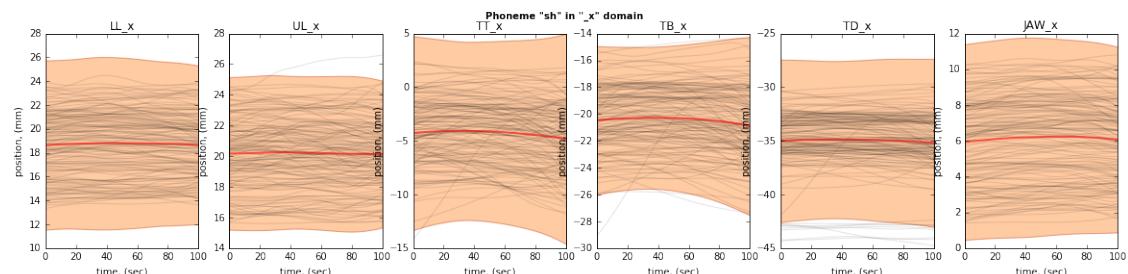
p



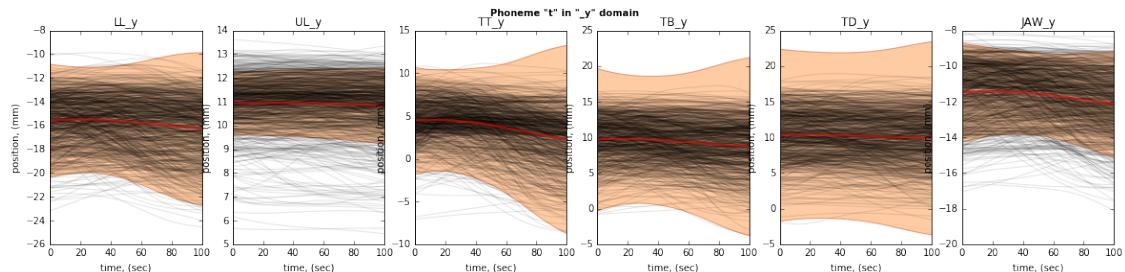
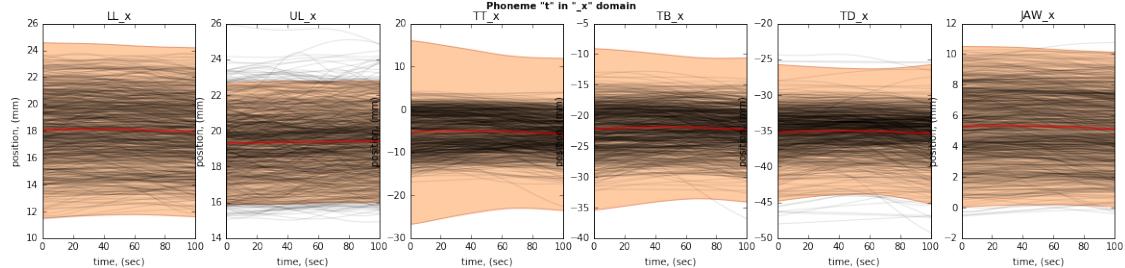
S



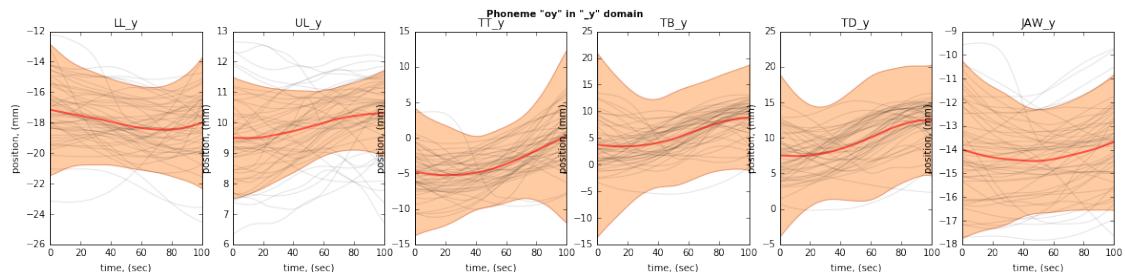
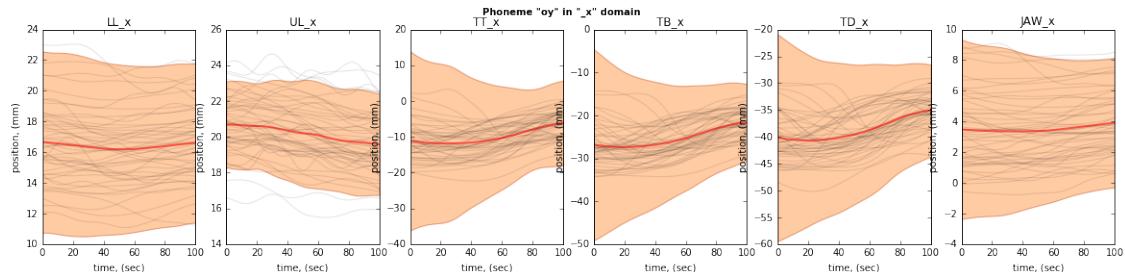
sh



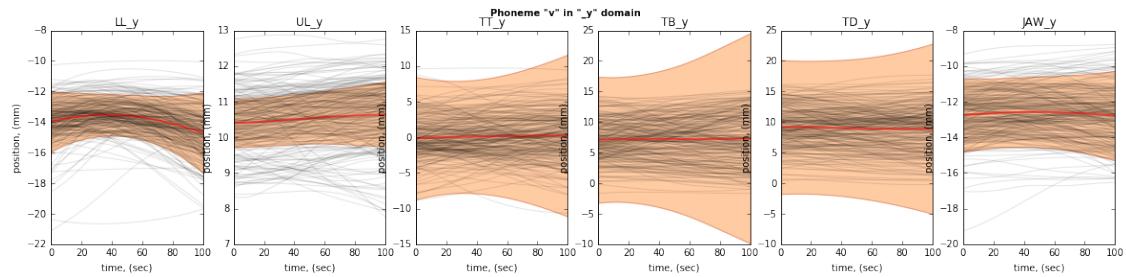
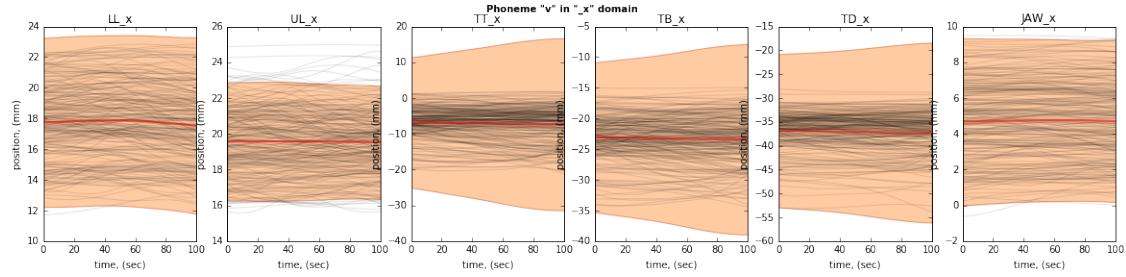
t



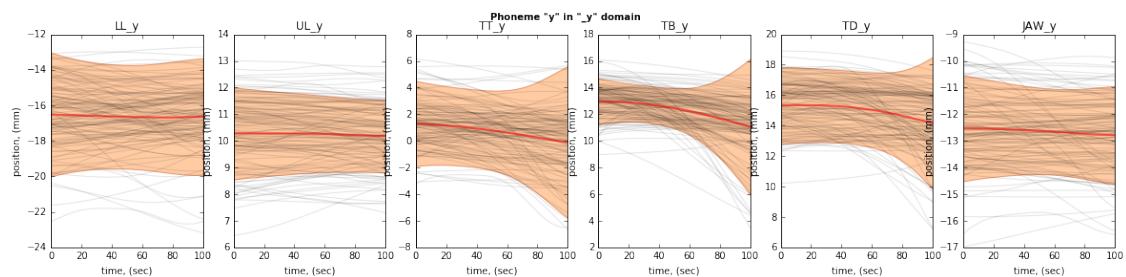
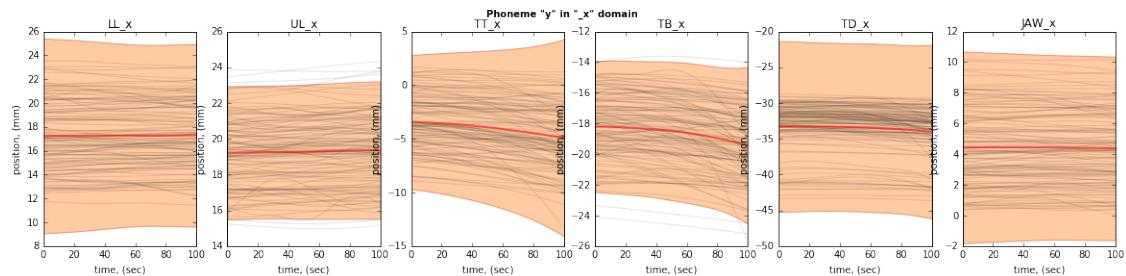
oy



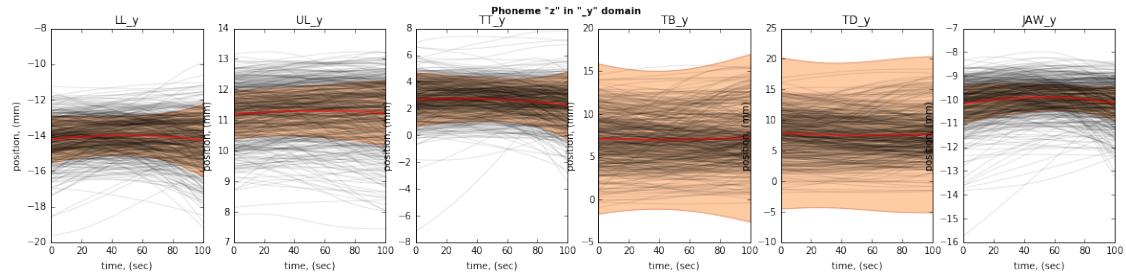
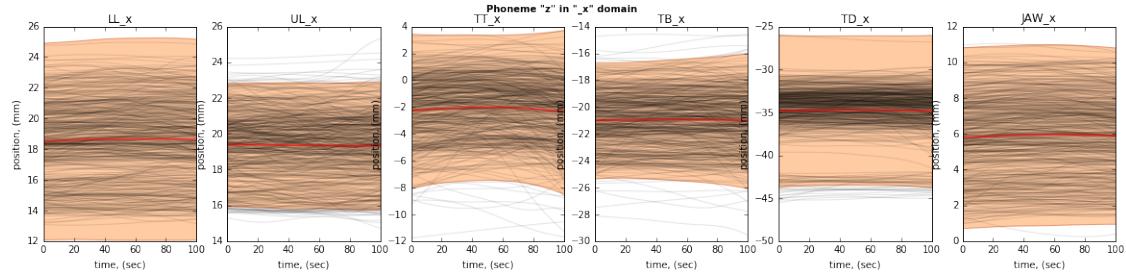
V



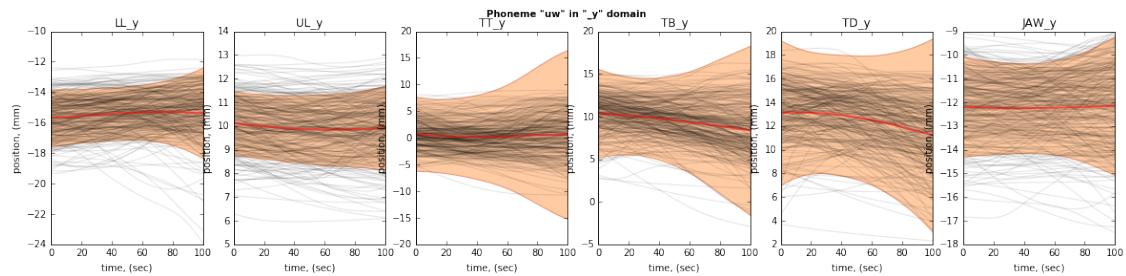
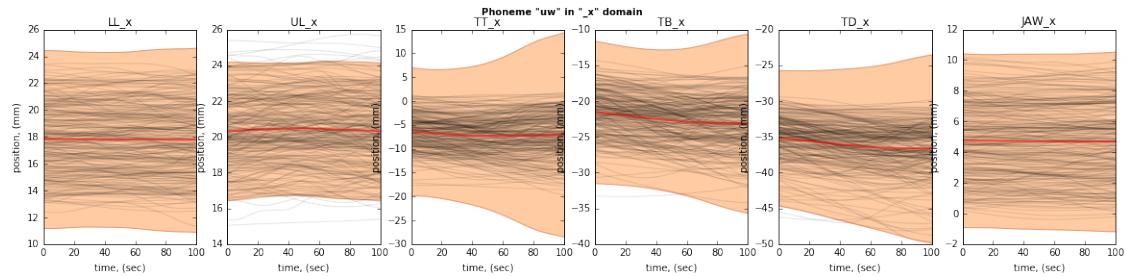
y



Z



uw



In []: