SwarmExpAnalysis

February 24, 2021

Paper Experiments Analysis

[18]:
```
import pandas as pd
from pandas import DataFrame, read_csv
import matplotlib.pyplot as plt
#SMALL_SIZE = 8
#MEDIUM_SIZE = 10
#BIGGER_SIZE = 36
#Font size modifiers for inclusion in papers.
#plt.rc('font', size=BIGGER_SIZE)          # controls default text sizes
#plt.rc('axes', titlesize=BIGGER_SIZE)     # fontsize of the axes title
#plt.rc('axes', labelsize=BIGGER_SIZE)    # fontsize of the x and y labels
#plt.rc('xtick', labelsize=BIGGER_SIZE)    # fontsize of the tick labels
#plt.rc('ytick', labelsize=BIGGER_SIZE)    # fontsize of the tick labels
#plt.rc('legend', fontsize=BIGGER_SIZE)    # legend fontsize
#plt.rc('figure', titlesize=BIGGER_SIZE)  # fontsize of the figure title
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 120)
%matplotlib inline
```

[19]:
```
# load all particle data from simulator
base = pd.read_csv('csv/base/exp.p.csv')
gap1 = pd.read_csv('csv/gap1/exp.p.csv')
exp1 = pd.read_csv('csv/exp1/exp.p.csv')
exp2 = pd.read_csv('csv/exp2/exp.p.csv')
exp3 = pd.read_csv('csv/exp3/exp.p.csv')
exp4 = pd.read_csv('csv/exp4/exp.p.csv')
exp5 = pd.read_csv('csv/exp5/exp.p.csv')
exp6 = pd.read_csv('csv/exp6/exp.p.csv')
gap1c = pd.read_csv('csv/gap1/exp.c.csv')
gap1r = pd.read_csv('csv/gap1/exp.r.csv')
basec = pd.read_csv('csv/base/exp.c.csv')
baser = pd.read_csv('csv/base/exp.r.csv')
exp1c = pd.read_csv('csv/exp1/exp.c.csv')
exp1r = pd.read_csv('csv/exp1/exp.r.csv')
exp2c = pd.read_csv('csv/exp2/exp.c.csv')
exp2r = pd.read_csv('csv/exp2/exp.r.csv')
exp3c = pd.read_csv('csv/exp3/exp.c.csv')
```

```
exp3r = pd.read_csv('csv/exp3/exp.r.csv')
exp4c = pd.read_csv('csv/exp4/exp.c.csv')
exp4r = pd.read_csv('csv/exp4/exp.r.csv')
exp5c = pd.read_csv('csv/exp5/exp.c.csv')
exp5r = pd.read_csv('csv/exp5/exp.r.csv')
exp6c = pd.read_csv('csv/exp6/exp.c.csv')
exp6r = pd.read_csv('csv/exp6/exp.r.csv')
LIMIT = min(gap1["STEP"].max(), base["STEP"].max(), exp1["STEP"].max(),␣
 ↪exp2["STEP"].max(), exp3["STEP"].max(), exp4["STEP"].max(), exp5["STEP"].
 ↪max(), exp6["STEP"].max())
base = base.loc[base['STEP']<=LIMIT]
gap1 = gap1.loc[gap1['STEP']<=LIMIT]
exp1 = exp1.loc[exp1['STEP']<=LIMIT]
exp2 = exp2.loc[exp2['STEP']<=LIMIT]
exp3 = exp3.loc[exp3['STEP']<=LIMIT]
exp4 = exp4.loc[exp4['STEP']<=LIMIT]
exp5 = exp5.loc[exp5['STEP']<=LIMIT]
exp6 = exp6.loc[exp6['STEP']<=LIMIT]
gap1c = gap1c.loc[gap1c['STEP']<=LIMIT]
gap1r = gap1r.loc[gap1r['STEP']<=LIMIT]
basec = basec.loc[basec['STEP']<=LIMIT]
baser = baser.loc[baser['STEP']<=LIMIT]
exp1c = exp1c.loc[exp1c['STEP']<=LIMIT]
exp1r = exp1r.loc[exp1r['STEP']<=LIMIT]
exp2c = exp2c.loc[exp2c['STEP']<=LIMIT]
exp2r = exp2r.loc[exp2r['STEP']<=LIMIT]
exp3c = exp3c.loc[exp3c['STEP']<=LIMIT]
exp3r = exp3r.loc[exp3r['STEP']<=LIMIT]
exp4c = exp4c.loc[exp4c['STEP']<=LIMIT]
exp4r = exp4r.loc[exp4r['STEP']<=LIMIT]
exp5c = exp5c.loc[exp5c['STEP']<=LIMIT]
exp5r = exp5r.loc[exp5r['STEP']<=LIMIT]
exp6c = exp6c.loc[exp6c['STEP']<=LIMIT]
exp6r = exp6r.loc[exp6r['STEP']<=LIMIT]
print("LIMIT:%d" % LIMIT)
```

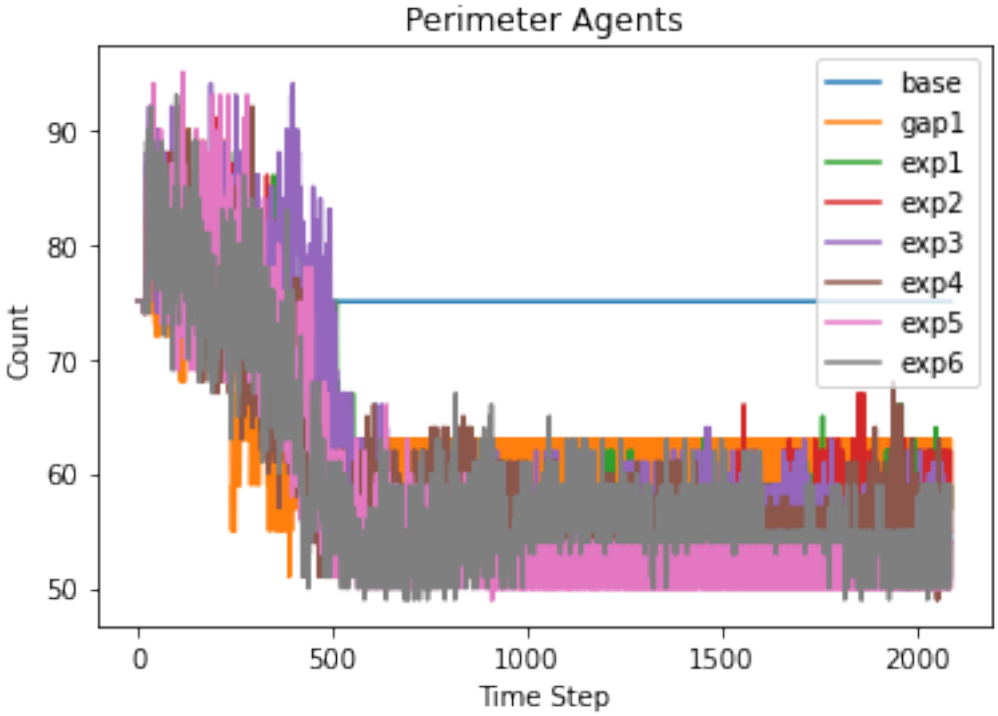
```
LIMIT:2086
```

[20]:
```
# generate a DataFrame with perimeter count per step
perimbase = base[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimgap1 = gap1[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimexp1 = exp1[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimexp2 = exp2[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
```

```
perimexp3 = exp3[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimexp4 = exp4[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimexp5 = exp5[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
perimexp6 = exp6[["STEP","PERIM"]].query("PERIM == True").groupby('STEP').
 ↪count().reset_index()
```

PERIMETER

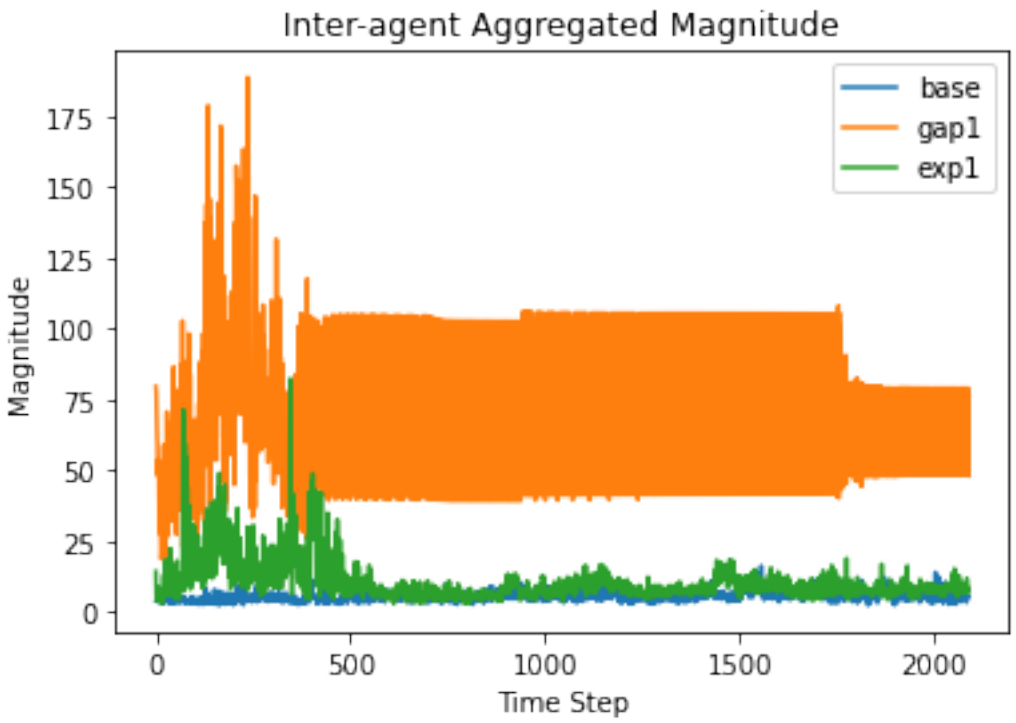
[21]:
```
%matplotlib inline
plt.plot(perimbase.STEP,perimbase.PERIM,label="base")
plt.plot(perimgap1.STEP,perimgap1.PERIM,label="gap1")
plt.plot(perimexp1.STEP,perimexp1.PERIM,label="exp1")
plt.plot(perimexp2.STEP,perimexp2.PERIM,label="exp2")
plt.plot(perimexp3.STEP,perimexp3.PERIM,label="exp3")
plt.plot(perimexp4.STEP,perimexp4.PERIM,label="exp4")
plt.plot(perimexp5.STEP,perimexp5.PERIM,label="exp5")
plt.plot(perimexp6.STEP,perimexp6.PERIM,label="exp6")
plt.title("Perimeter Agents")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Count")
plt.show()
```



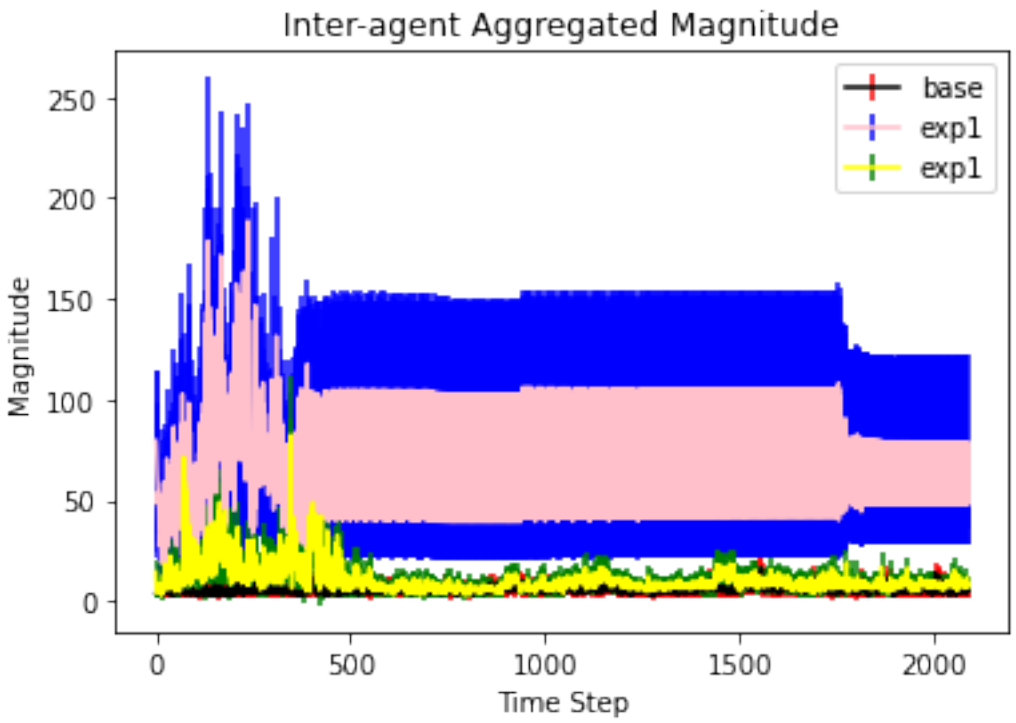
AGGREGATED MAGNITUDE DATA

[22]:
```
basemean = base[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
basestd = base[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
gap1mean = gap1[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
gap1std = gap1[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp1mean = exp1[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp1std = exp1[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp2mean = exp2[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp2std = exp2[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp3mean = exp3[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp3std = exp3[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp4mean = exp4[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp4std = exp4[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp5mean = exp5[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp5std = exp5[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
exp6mean = exp6[["STEP","IMAG"]].groupby('STEP').mean(0).reset_index()
exp6std = exp6[["STEP","IMAG"]].groupby('STEP').std(0).reset_index()
```

[23]:
```
%matplotlib inline
plt.plot(basemean.STEP,basemean.IMAG,label='base')
plt.plot(gap1mean.STEP,gap1mean.IMAG,label='gap1')
plt.plot(exp1mean.STEP,exp1mean.IMAG,label='exp1')
#plt.plot(exp2mean.STEP,exp2mean.IMAG,label='exp2')
#plt.plot(exp3mean.STEP,exp3mean.IMAG,label='exp3')
#plt.plot(exp4mean.STEP,exp4mean.IMAG,label='exp4')
#plt.plot(exp5mean.STEP,exp5mean.IMAG,label='exp5')
#plt.plot(exp6mean.STEP,exp6mean.IMAG,label='exp6')
plt.title("Inter-agent Aggregated Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Magnitude")
plt.show()
```



[24]:
```
plt.errorbar(basemean.STEP,basemean.IMAG,basestd.
 ↪IMAG,label='base',ecolor='red',color='black')
plt.errorbar(gap1mean.STEP,gap1mean.IMAG,gap1std.
 ↪IMAG,label='exp1',ecolor='blue',color='pink')
plt.errorbar(exp1mean.STEP,exp1mean.IMAG,exp1std.
 ↪IMAG,label='exp1',ecolor='green',color='yellow')
plt.title("Inter-agent Aggregated Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Magnitude")
plt.show()
```

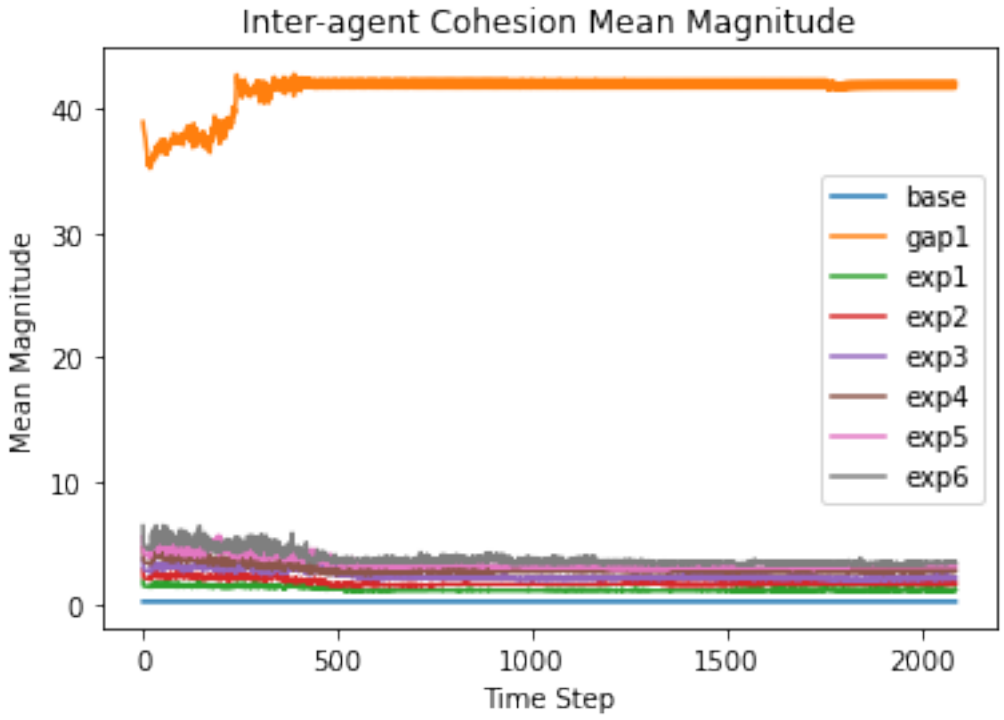


COHESION

[25]:
```
basecmean = basec[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
basecstd = basec[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
gap1cmean = gap1c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
gap1cstd = gap1c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp1cmean = exp1c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp1cstd = exp1c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp2cmean = exp2c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp2cstd = exp2c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp3cmean = exp3c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp3cstd = exp3c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp4cmean = exp4c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp4cstd = exp4c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp5cmean = exp5c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp5cstd = exp5c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp6cmean = exp6c[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp6cstd = exp6c[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
```

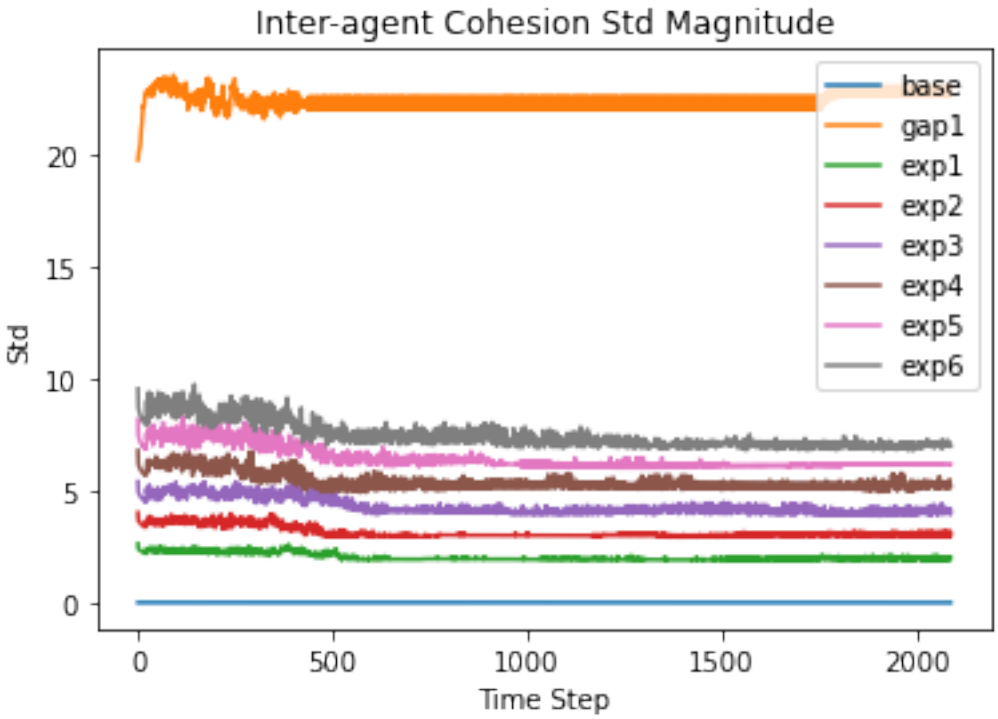
[26]:
```
%matplotlib inline
plt.plot(basecmean.STEP,basecmean.MAG,label='base')
plt.plot(gap1cmean.STEP,gap1cmean.MAG,label='gap1')
plt.plot(exp1cmean.STEP,exp1cmean.MAG,label='exp1')
plt.plot(exp2cmean.STEP,exp2cmean.MAG,label='exp2')
```

```
plt.plot(exp3cmean.STEP,exp3cmean.MAG,label='exp3')
plt.plot(exp4cmean.STEP,exp4cmean.MAG,label='exp4')
plt.plot(exp5cmean.STEP,exp5cmean.MAG,label='exp5')
plt.plot(exp6cmean.STEP,exp6cmean.MAG,label='exp6')
plt.title("Inter-agent Cohesion Mean Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Mean Magnitude")
plt.show()
```

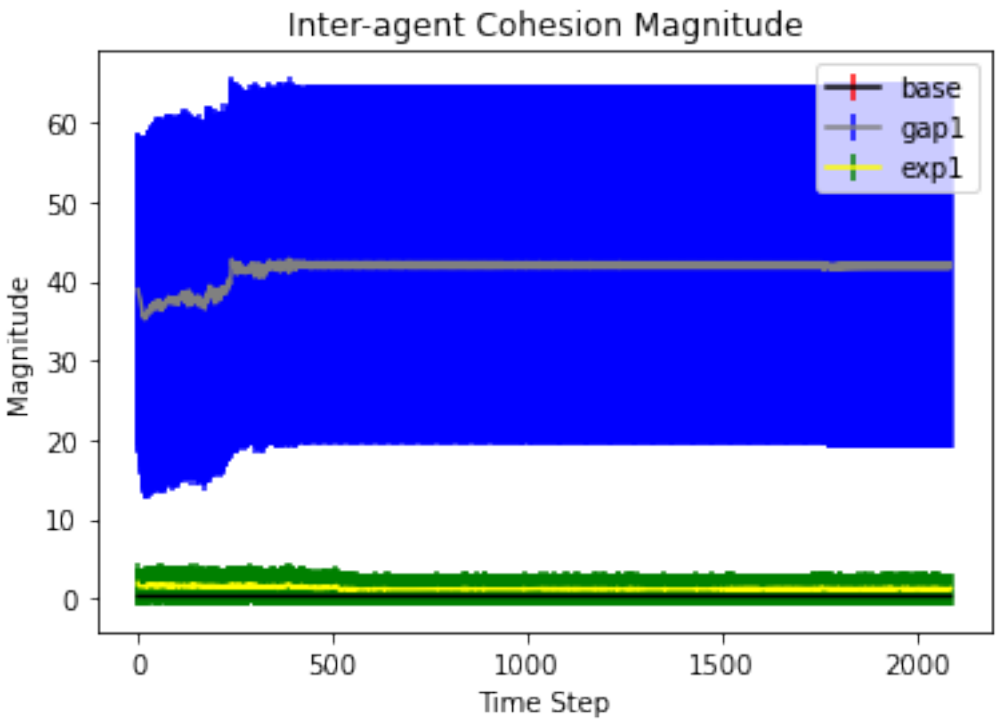


[27]:
```
%matplotlib inline
plt.plot(basecstd.STEP,basecstd.MAG,label='base')
plt.plot(gap1cstd.STEP,gap1cstd.MAG,label='gap1')
plt.plot(exp1cstd.STEP,exp1cstd.MAG,label='exp1')
plt.plot(exp2cstd.STEP,exp2cstd.MAG,label='exp2')
plt.plot(exp3cstd.STEP,exp3cstd.MAG,label='exp3')
plt.plot(exp4cstd.STEP,exp4cstd.MAG,label='exp4')
plt.plot(exp5cstd.STEP,exp5cstd.MAG,label='exp5')
plt.plot(exp6cstd.STEP,exp6cstd.MAG,label='exp6')
plt.title("Inter-agent Cohesion Std Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Std")
```

```
plt.show()
```



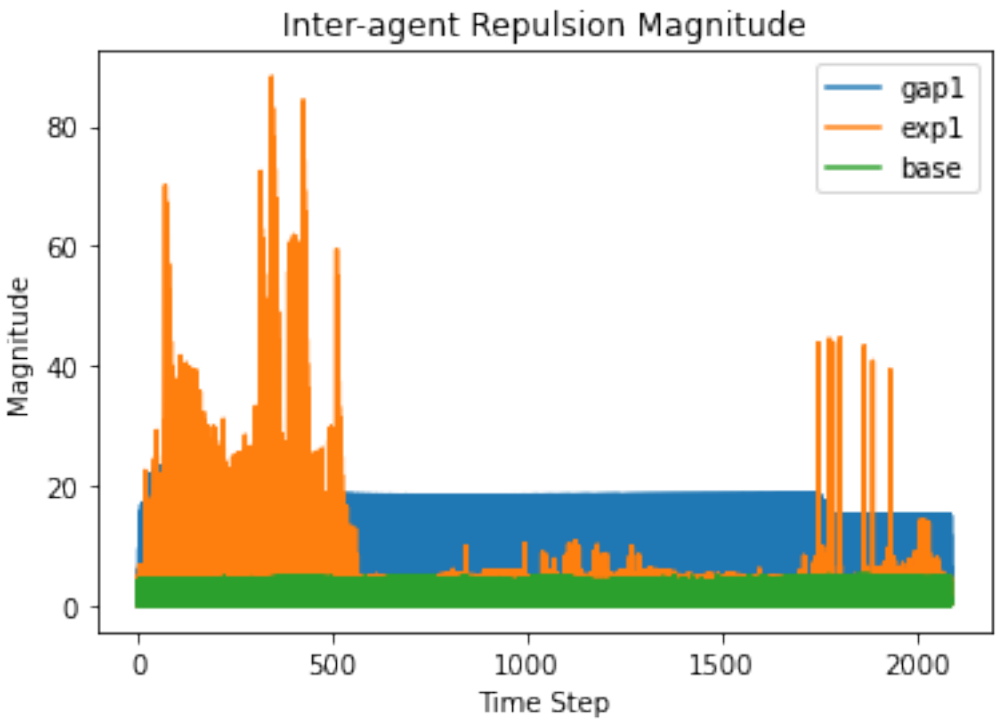
[28]:
```
%matplotlib inline
plt.errorbar(basecmean.STEP,basecmean.MAG,basecstd.
 ↪MAG,label='base',ecolor='red',color='black')
plt.errorbar(gap1cmean.STEP,gap1cmean.MAG,gap1cstd.
 ↪MAG,label='gap1',ecolor='blue',color='grey')
plt.errorbar(exp1cmean.STEP,exp1cmean.MAG,exp1cstd.
 ↪MAG,label='exp1',ecolor='green',color='yellow')
plt.title("Inter-agent Cohesion Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Magnitude")
plt.show()
```



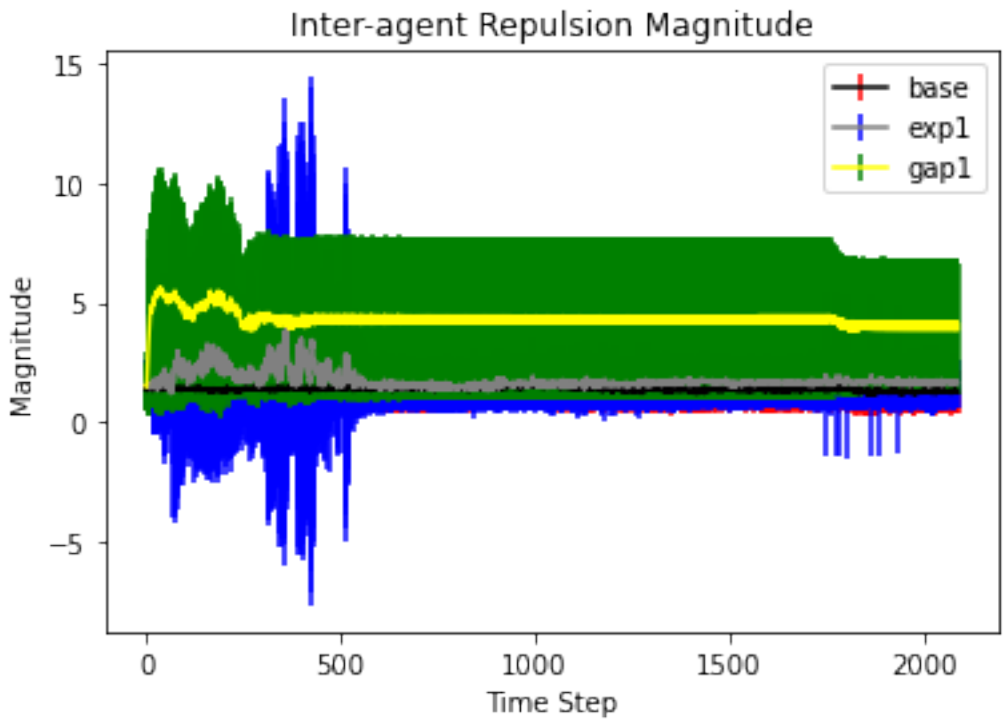
REPULSION

[29]:
```
basermean = baser[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
baserstd = baser[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
gap1rmean = gap1r[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
gap1rstd = gap1r[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp1rmean = exp1r[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp1rstd = exp1r[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
exp2rmean = exp2r[["STEP","MAG"]].groupby('STEP').mean(0).reset_index()
exp2rstd = exp2r[["STEP","MAG"]].groupby('STEP').std(0).reset_index()
```

[30]:
```
%matplotlib inline
plt.plot(gap1r.STEP,gap1r.MAG,label='gap1')
plt.plot(exp1r.STEP,exp1r.MAG,label='exp1')
plt.plot(baser.STEP,baser.MAG,label='base')
plt.title("Inter-agent Repulsion Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Magnitude")
plt.show()
```



[31]:
```
%matplotlib inline
plt.errorbar(basermean.STEP,basermean.MAG,baserstd.
 ↪MAG,label='base',ecolor='red',color='black')
plt.errorbar(exp1rmean.STEP,exp1rmean.MAG,exp1rstd.
 ↪MAG,label='exp1',ecolor='blue',color='grey')
plt.errorbar(gap1rmean.STEP,gap1rmean.MAG,gap1rstd.
 ↪MAG,label='gap1',ecolor='green',color='yellow')
plt.title("Inter-agent Repulsion Magnitude")
plt.legend()
plt.xlabel("Time Step")
plt.ylabel("Magnitude")
plt.show()
```

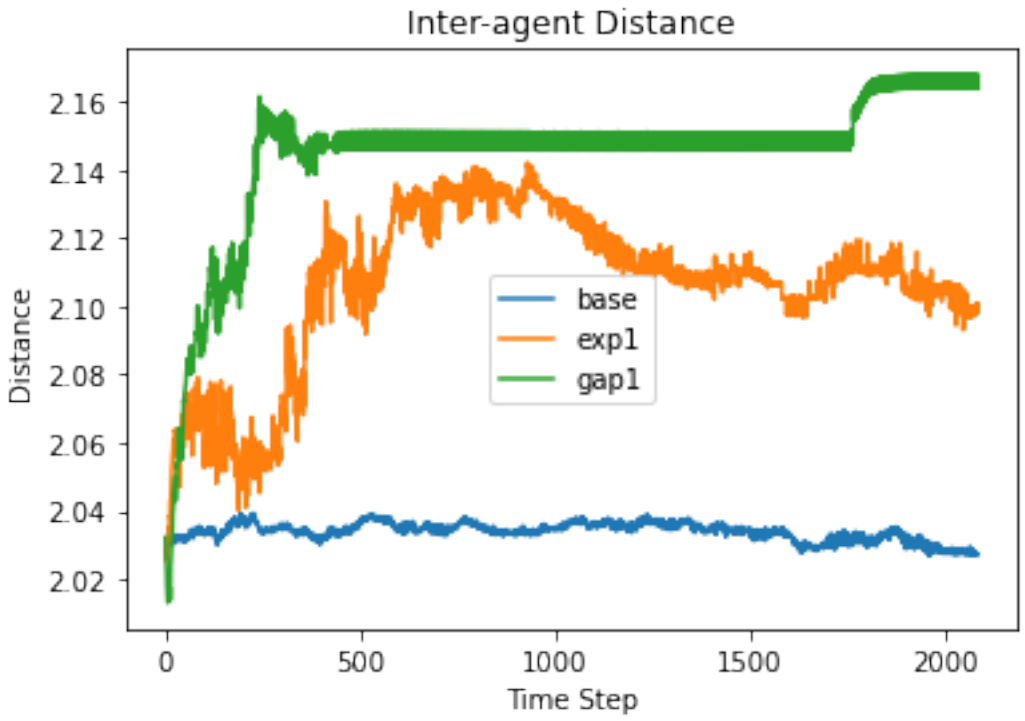


DISTANCE

[32]:
```
basedmean = basec[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
basedstd = basec[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp1dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp1dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp2dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp2dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp3dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp3dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp4dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp4dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp5dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp5dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
exp6dmean = exp1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
exp6dstd = exp1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
gap1dmean = gap1c[["STEP","DIST"]].groupby('STEP').mean(0).reset_index()
gap1dstd = gap1c[["STEP","DIST"]].groupby('STEP').std(0).reset_index()
```

[33]:
```
%matplotlib inline
plt.plot(basedmean.STEP,basedmean.DIST,label='base')
plt.plot(exp1dmean.STEP,exp1dmean.DIST,label='exp1')
plt.plot(gap1dmean.STEP,gap1dmean.DIST,label='gap1')
plt.title("Inter-agent Distance")
```

```
plt.xlabel("Time Step")
plt.legend()
plt.ylabel("Distance")
plt.show()
```



[34]:
```
%matplotlib inline
plt.errorbar(gap1dmean.STEP,gap1dmean.DIST,gap1dstd.
 ↪DIST,label='gap1',ecolor='green',color='yellow')
plt.errorbar(exp3dmean.STEP,exp3dmean.DIST,exp3dstd.
 ↪DIST,label='exp6',ecolor='red',color='black')
plt.errorbar(basedmean.STEP,basedmean.DIST,basedstd.
 ↪DIST,label='Base',ecolor='blue',color='grey')
plt.legend()
plt.title("Inter-agent Distance")
plt.xlabel("Time Step")
plt.ylabel("Distance")
plt.show()
```


Inter-agent Distance
gap1
exp6
Base
Distance
2.5
2.4
2.3
2.2
2.1
2.0
1.9
1.8
0
500
1000
1500
2000
Time Step

[ ]: