```
In [8]: T = time.time()
        correlation = 200
        position = np.nan*np.zeros_like(prices.values)

        returns_adj = prices.apply(returns_adjust, com=32, clip=4.2)
        # DCC by Engle
        cor = returns_adj.ewm(com=correlation, min_periods=correlation).corr()

        mu = np.tanh(returns_adj.cumsum().apply(osc)).values
        vo = prices.pct_change().ewm(com=32, min_periods=32).std().values

        for n,t in enumerate(prices.index):
            matrix = shrink2id(cor.loc[t].values, lamb=0.5)
            risk_position = solve(matrix, mu[n])/a_norm(mu[n], matrix)
            position[n] = risk_position/vo[n]

        portfolio = Portfolio(prices, pd.DataFrame(index=prices.index, columns=prices.keys()
        print(time.time()-T)
```

15.90978717803955

```
In [9]: analysis(portfolio.nav())
```

Conclusions

- Dramatic relativ improvements observable despite using the same signals as in previous Experiment.
- Main difference here is to take advantage of cross-correlations in the risk measurement.
- Possible to add constraints on individual assets or groups of them.
- Possible to reflect trading costs in objective with regularization terms (Ridge, Lars, Elastic Nets, …)