Bertram Solutions LLC
1525 4th Street
Kirkland, WA 98033
(425) 616-2755
http://www.bertram.solutions
info@bertram.solutions

Travis Cook

RIDGELINE ASSET MANAGEMENT
5 Cowboys Way, Suite 300
Frisco, TX 75034

tcook@ridgelineam.com

Kirkland, March 22nd, 2022

Hello Travis,

Please find some thoughts and observations regarding momentum strategies.

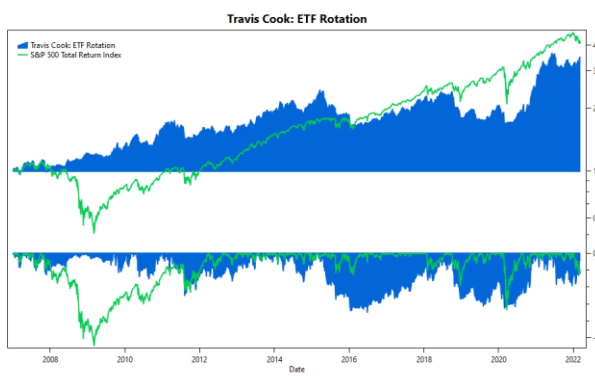
Thank you,
Best regards.

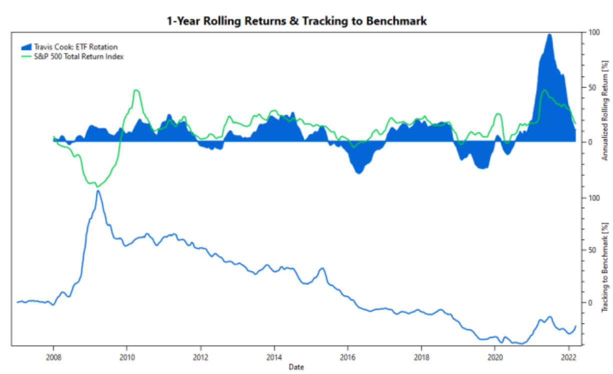
Felix Bertram

ETF Rotation

We ported the strategy to TuringTrader. As part of this effort, we clarified the purpose of the *rankExit* parameter. We summarize the strategy's rules as follows:

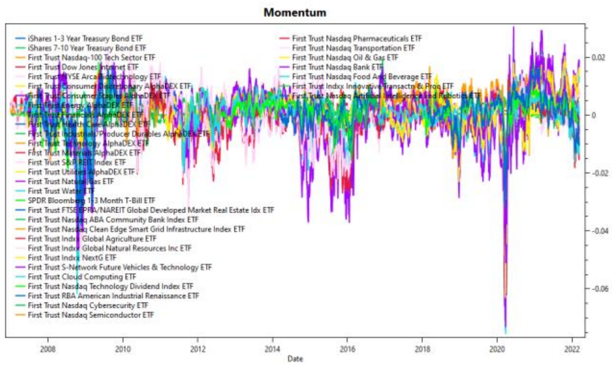
- Trade a universe of ETFs representing industry sectors, sub-sectors, and bonds
- Rebalance on a monthly schedule, ignore all inter-month data and signals
- Only hold positions while the S&P 500 closes above its 200-day (9.5 months) moving average
- Rank assets by their momentum, calculated as the average of the asset's 1, 3, 6, and 12-months rate of change
- Hold any asset, while its momentum is in the top-10
- Buy assets with top momentum rank, for a total of two assets held
- Allocate 50% of the available capital to each

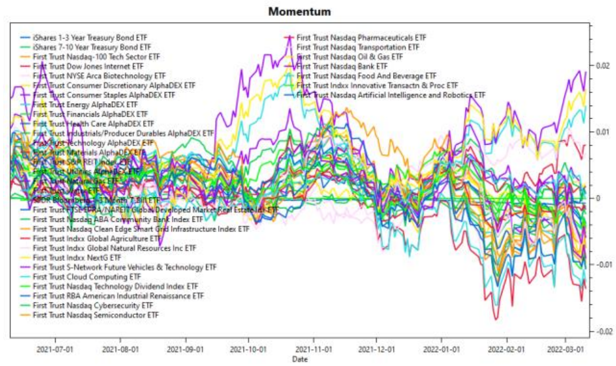




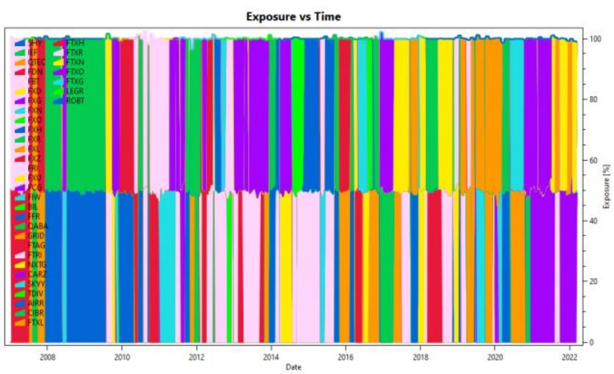
Metric		Travis Cook: ETF Rotation	S&P 500 Total Return Index
Simulation Start	01/03/2007	$1,000.00	$1,000.00
Simulation End	03/10/2022	$3,497.94	$4,098.55
Simulation Period	15.2 years		
Compound Annual Growth Rate		8.60%	9.74%
Stdev of Returns (Monthly, Annualized)		17.38%	18.24%
Maximum Drawdown (Daily)		35.19%	55.25%
Maximum Flat Days		2087.00 days	1637.00 days
Sharpe Ratio (Rf=T-Bill, Monthly, Annualizec		0.42	0.47
Beta (To Benchmark, Monthly)		0.38	- benchmark -
Ulcer Index		15.09%	13.87%
Ulcer Performance Index (Martin Ratio)		0.57	0.70

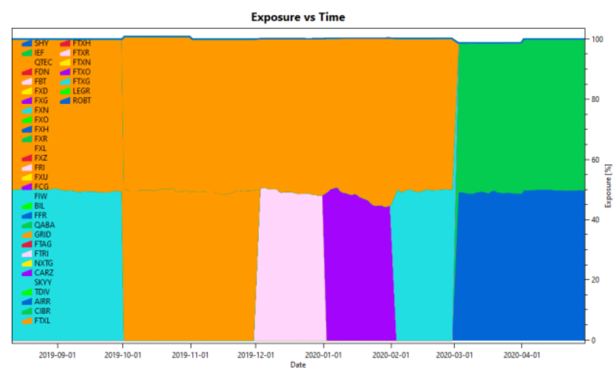
The performance looks very similar to the Amibroker backtest in Modification #2. The shape of the equity curve is different, because TuringTrader's plot is logarithmic, while the plot in the previous report was linear.





Looking at the momentum signal shows much of the same issues we have seen during our research of bond strategies. The momentum signal is very noisy. This noise will inevitably find its way into the ranking, and negatively affect the validity of trading decisions.





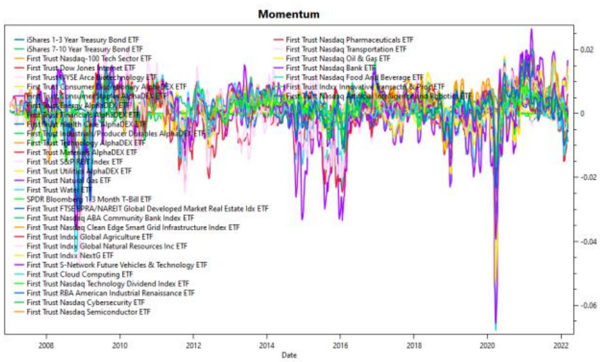
The exposure chart shows how the strategy is constantly rotating its assets, often every month. This quick rotation is evidence that an asset’s ranking may easily change by ten or more positions within just a month.

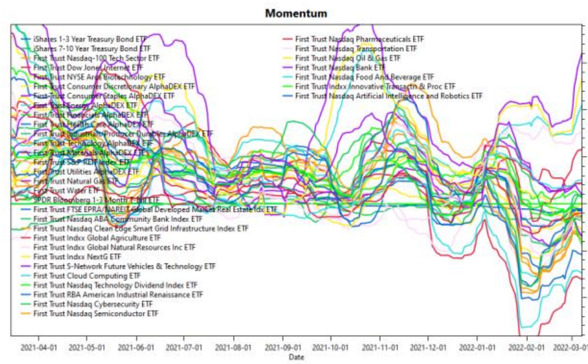
Modification #1: Pre-Filtered 1-3-6-12 Momentum

The first modification we make is to calculate momentum as follows:

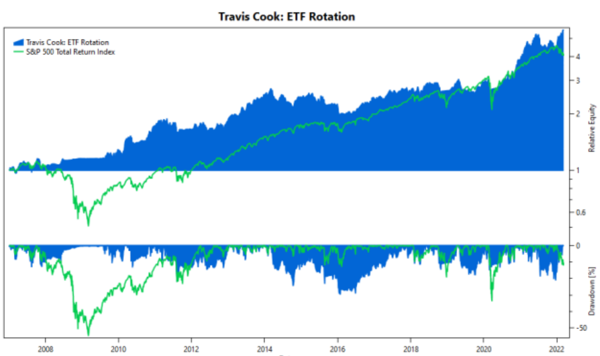
```
var series = i.TypicalPrice().KAMA();
var mom = series.Momentum(1 * 21)[0]
    + series.Momentum(3 * 21)[0]
    + series.Momentum(6 * 21)[0]
    + series.Momentum(12 * 21)[0];
```

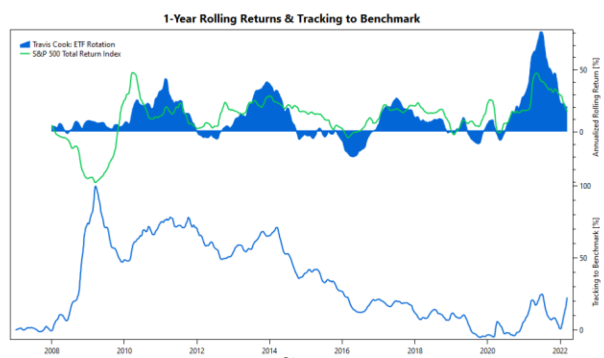
By using the typical price and applying Kaufman's Adaptive Moving Average (KAMA), we hope to better eliminate noise.





The screenshots show how the momentum signals are significantly less noisy than they were before.





Metric		Travis Cook: ETF Rotation	S&P 500 Total Return Index
Simulation Start	01/03/2007	$1,000.00	$1,000.00
Simulation End	03/10/2022	$5,556.22	$4,098.55
Simulation Period	15.2 years		
Compound Annual Growth Rate		11.96%	9.74%
Stdev of Returns (Monthly, Annualized)		16.83%	18.24%
Maximum Drawdown (Daily)		29.80%	55.25%
Maximum Flat Days		1357.00 days	1637.00 days
Sharpe Ratio (Rf=T-Bill, Monthly, Annualizec		0.62	0.47
Beta (To Benchmark, Monthly)		0.35	- benchmark -
Ulcer Index		10.09%	13.87%
Ulcer Performance Index (Martin Ratio)		1.19	0.70

We see that the strategy has significantly improved. This supports our statement that the noisy momentum signal is problematic. However, even with these changes, the strategy is still significantly underperforming its benchmark.

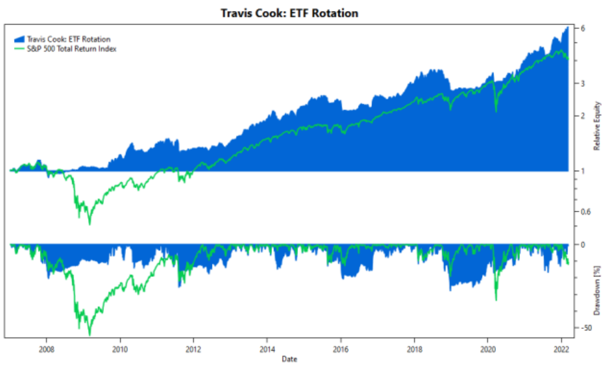
Modification #2: Log-Regression Momentum

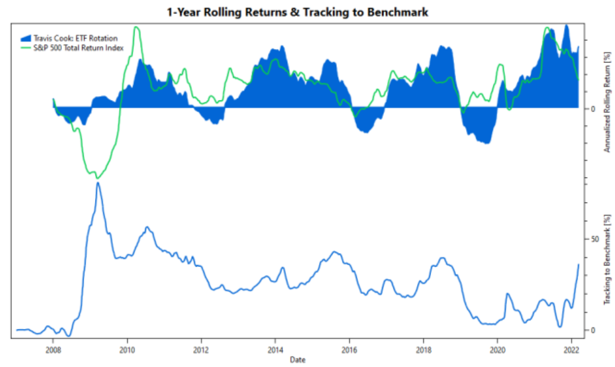
We modified the momentum calculation as follows:

```
var series = i.TypicalPrice();
var mom = 100.0 * 252.0 * series.LogRegression(MOM_PER).Slope[0];
```

Further, we optimized the strategy for CAR/MDD, resulting in the following settings:

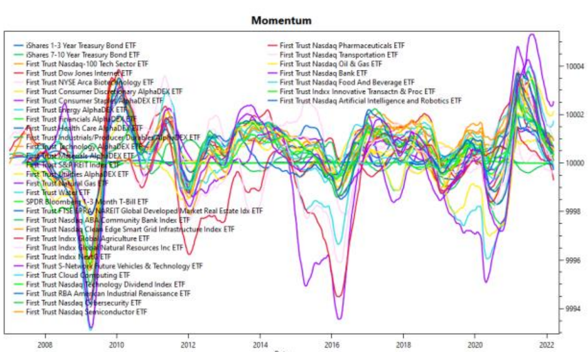
```
MOM_PER = 252
NUM_POS = 3
RANK_EXIT = 5
```

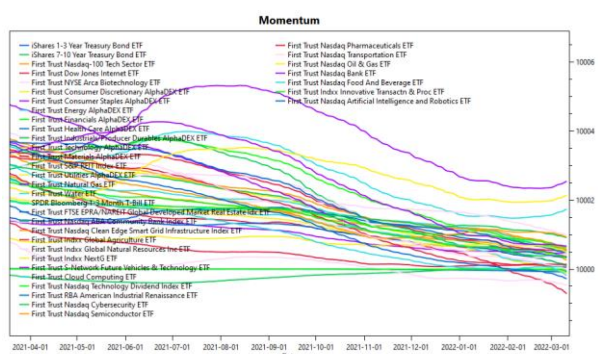




Metric		Travis Cook: ETF Rotation	S&P 500 Total Return Index
Simulation Start	01/03/2007	$1,000.00	$1,000.00
Simulation End	03/10/2022	$6,119.23	$4,098.55
Simulation Period	15.2 years		
Compound Annual Growth Rate		12.67%	9.74%
Stdev of Returns (Monthly, Annualized)		16.09%	18.24%
Maximum Drawdown (Daily)		27.98%	55.25%
Maximum Flat Days		800.00 days	1637.00 days
Sharpe Ratio (Rf=T-Bill, Monthly, Annualizec		0.69	0.47
Beta (To Benchmark, Monthly)		0.35	- benchmark -
Ulcer Index		10.33%	13.87%
Ulcer Performance Index (Martin Ratio)		1.23	0.70

This is a significant step up. The strategy is now able to keep up with its benchmark, while at the same time showing lower volatility and less downside.





Again, we attribute these improvements to the less noisy momentum signals. However, we are not happy with the 12-months momentum period, which leaves the strategy vulnerable should market conditions change quickly. The rolling returns show this prolonged negative returns.

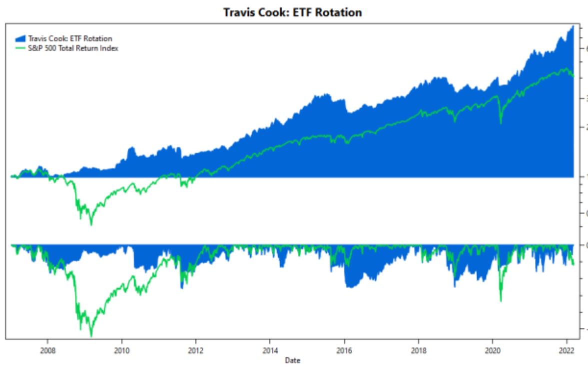
Modification #3: Log-Regression Pre-Filter + Low-Pass Momentum

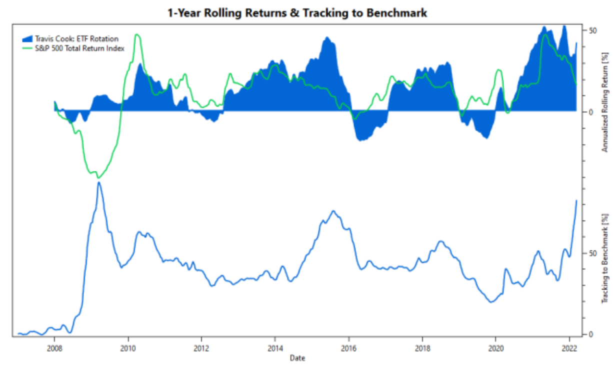
In an attempt to shorten the momentum periods, we tried the following formula:

```
var series = i.TypicalPrice().LinRegression(REGR_PER).Intercept;
var mom = series.LogReturn().EMA(MOM_PER).EMA(MOM_PER);
```

With the following parameters:

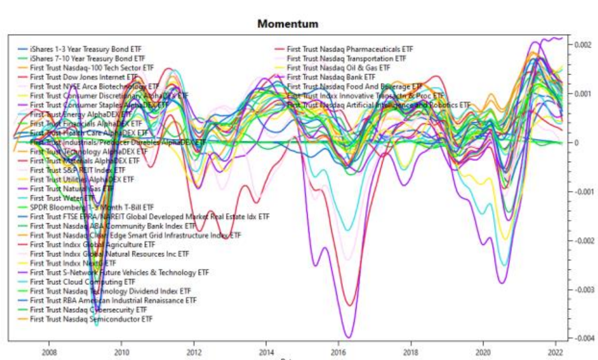
```
MOM_PER = 231
REGR_PER = 168
NUM_POS = 2
RANK_EXIT = 3
```

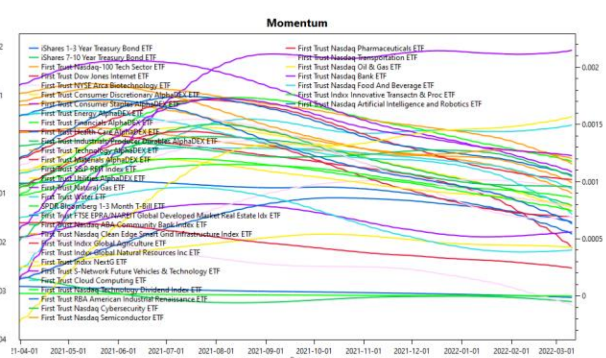




Metric		Travis Cook: ETF Rotation	S&P 500 Total Return Index
Simulation Start	01/03/2007	$1,000.00	$1,000.00
Simulation End	03/10/2022	$8,289.97	$4,098.55
Simulation Period	15.2 years		
Compound Annual Growth Rate		14.95%	9.74%
Stdev of Returns (Monthly, Annualized)		16.00%	18.24%
Maximum Drawdown (Daily)		26.46%	55.25%
Maximum Flat Days		816.00 days	1637.00 days
Sharpe Ratio (Rf=T-Bill, Monthly, Annualizec		0.82	0.47
Beta (To Benchmark, Monthly)		0.34	- benchmark -
Ulcer Index		9.81%	13.87%
Ulcer Performance Index (Martin Ratio)		1.52	0.70

The performance has further improved. However, the strategy is still too slow to respond to changing market conditions.





We notice that the momentum signals look quite different than before, but we are unsure why that is.

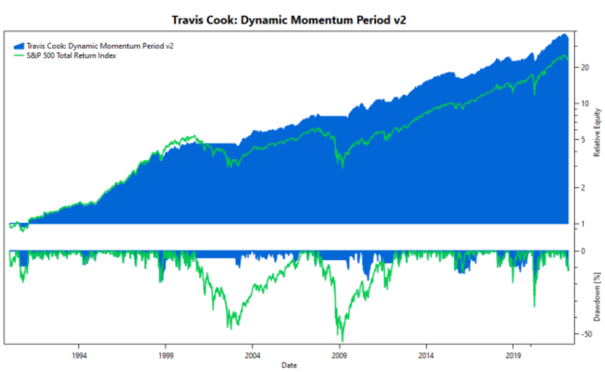
Dynamic Momentum Period: Based on R2 of Log-Regression

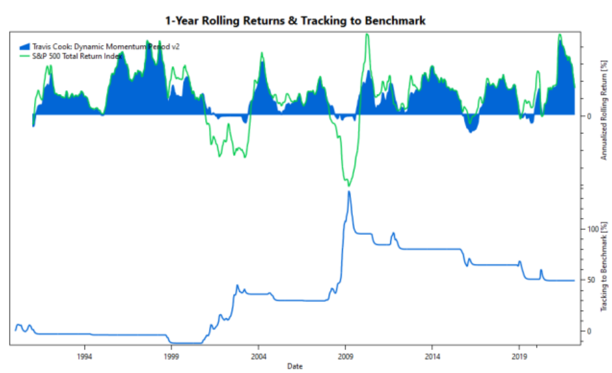
The previous tests have shown that we want long momentum periods for better noise suppression, but we need shorter momentum periods to make the strategies respond faster to market fluctuations. What we are looking for is a scheme to automatically adjust the momentum period such that we can combine the best of the two worlds.

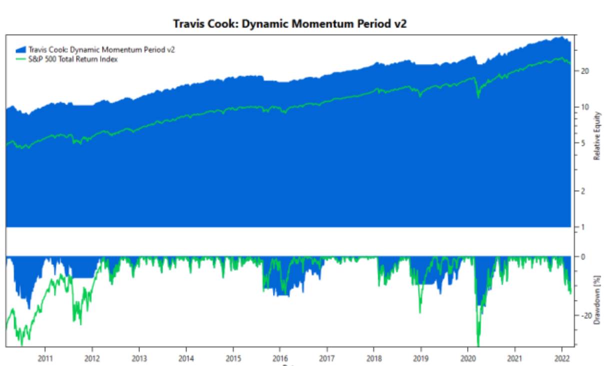
We can use linear or logarithmic regression to determine a best-fit curve through a noisy set of data. In addition to the slope, a regression provides a measure for the quality of the fit: R2. We experimented with dynamically adjusting the filter period for a momentum calculation as follows:

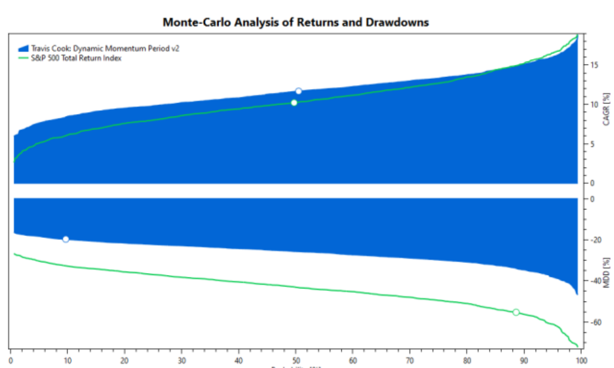
```
var assetFlt = asset.Instrument.TypicalPrice().KAMA();
var regr = assetFlt.LogRegression(REGR_PER);
var relR2 = Math.Min(1.0, Math.Max(0.0, (regr.R2[0] - R2_MIN / 100.0)
    / (R2_MAX / 100.0 - R2_MIN / 100.0)));
var period = FLT_MIN + Math.Pow(relR2, R2_POW / 100.0) * (FLT_MAX - FLT_MIN);
var sc = 2.0 / (1.0 + Math.Max(21, period));
var r = assetFlt.LogReturn()[0];
mom = sc * r + mom * (1 - sc);
```

We tested this method with a simple strategy holding SPX when the momentum is positive and exits to cash when the momentum is negative.

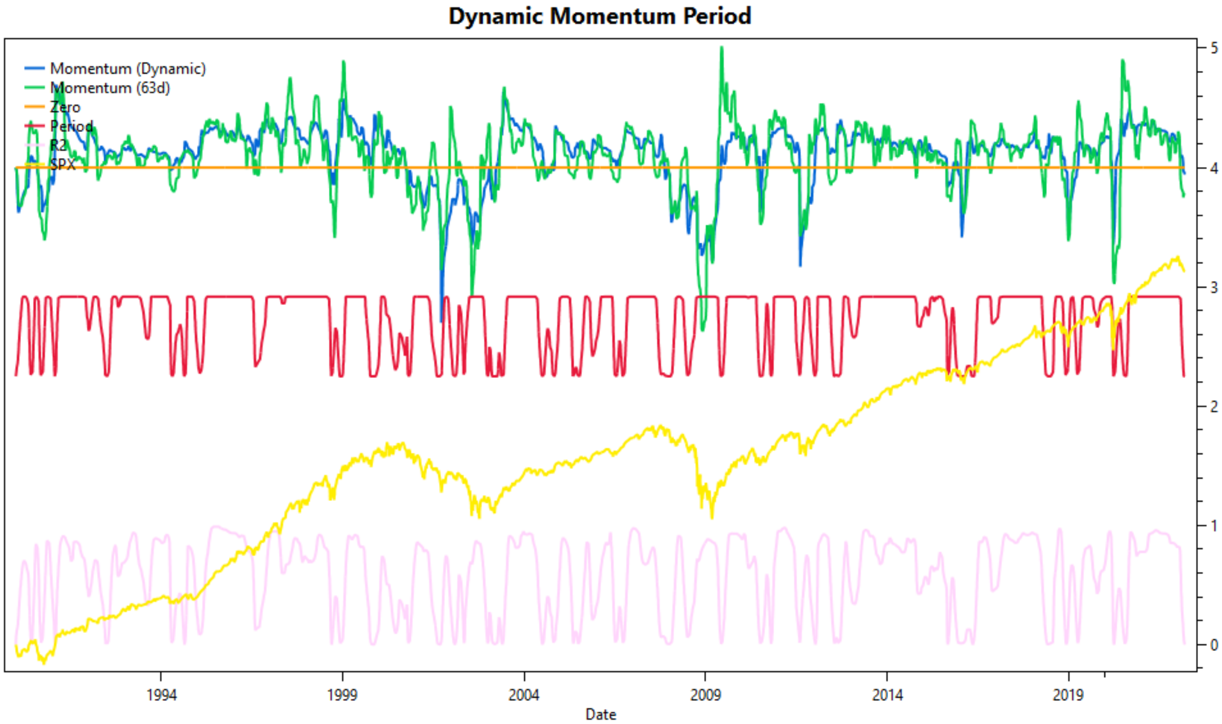








At a first glance we can see how the strategy did extremely well in the recessions of 2000 and 2008. The performance after 2008 might seem a bit underwhelming. However, zooming in reveals that this method did reduce risk in 2012, 2019, and 2020. The Monte-Carlo simulation confirms this statement.



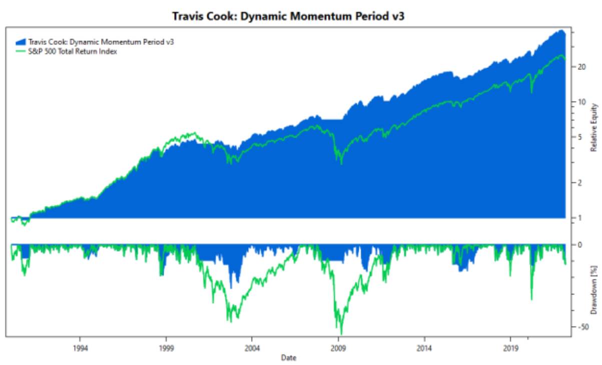
The momentum signal derived with this method (blue) is significantly cleaner than the momentum signal calculated with a static 63d filter. We are hopeful that this improved momentum signal can lead to better ranking quality when used in a momentum strategy.

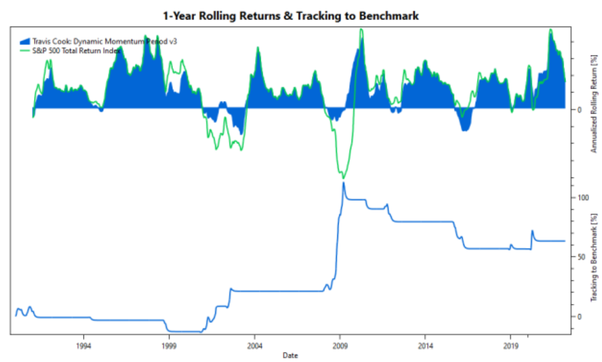
Dynamic Momentum Period: Based on Volatility

We tried a simplified version, scaling the filter period based on historical volatility. We are using the following approach:

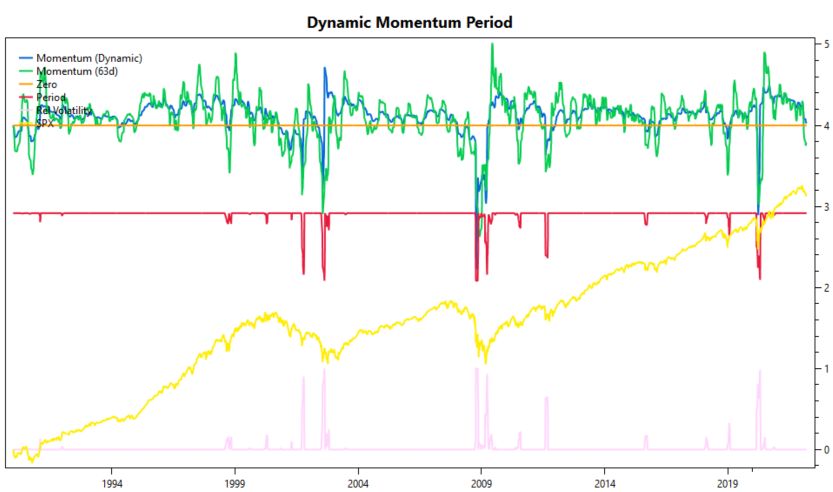
```
var assetFlt = asset.Instrument.TypicalPrice().KAMA();
var lr = assetFlt.LogReturn();
var vol = 100.0 * Math.Sqrt(252.0) * lr.StandardDeviation(VOL_PER)[0];
var relVol = Math.Min(1.0, Math.Max(0.0, (vol - VOL_MIN)
    / (VOL_MAX - VOL_MIN)));
var period = FLT_MAX - Math.Pow(relVol, VOL_POW / 100.0) * (FLT_MAX - FLT_MIN);
var sc = 2.0 / (1.0 + Math.Max(21, period));
mom = sc * lr[0] + mom * (1 - sc);
```

Again, we tested this momentum signal with a simple SPX trading strategy.





The results look surprisingly similar to the previous method. The rolling returns show some differences though: this new strategy has more pronounced dips into negative returns as the previous one. Most likely, this behavior indicates that this method is slower to respond.



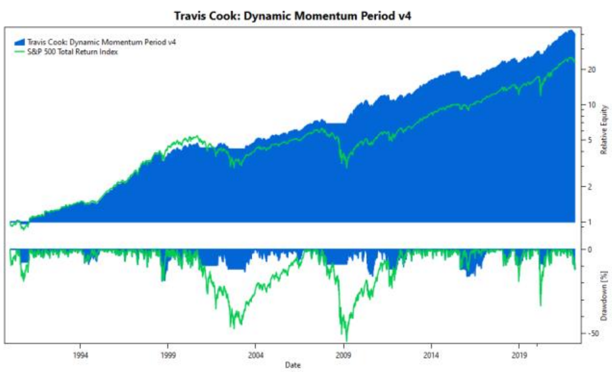
The differences only become more obvious, when we look at the momentum signal. Unlike the previous version, this simplified method stays at the same long-term filter for most of the time, only adjusting the momentum period downwards for very short periods of time. Generally, this is preferred over the previous method, as it helps keeping the noise low. However, as the slightly increased drawdowns show, this method might take it a bit too far.

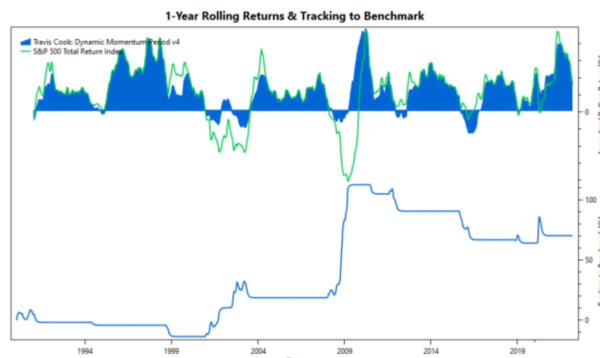
Dynamic Momentum Period: Based on Trading Range

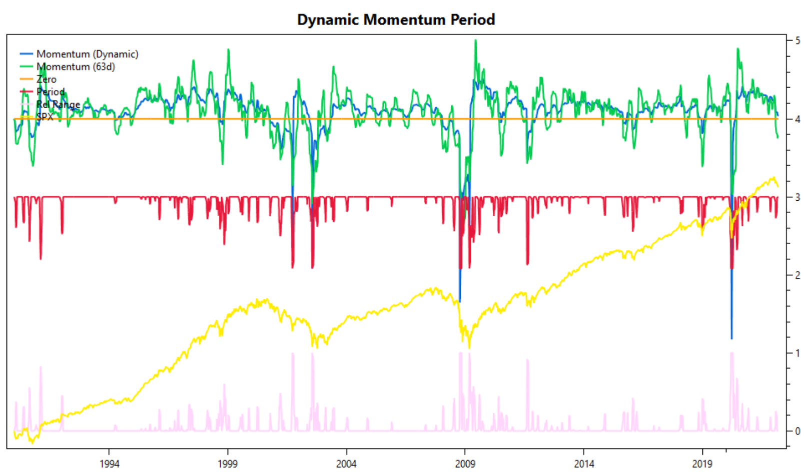
In another attempt to simplify the period calculation, we tried this code:

```
var assetFlt = asset.Instrument.TypicalPrice().KAMA();
var rng = (assetFlt.Highest(RNG_PER)[0] - assetFlt.Lowest(RNG_PER)[0])
          / assetFlt[0];
var relRng = Math.Min(1.0, Math.Max(0.0, (rng - RNG_MIN / 100.0)
          / (RNG_MAX / 100.0 - RNG_MIN / 100.0)));
var period = FLT_MIN + Math.Pow(1.0 - relRng, VOL_POW / 100.0)
          * (FLT_MAX - FLT_MIN);
var sc = 2.0 / (1.0 + Math.Max(21, period));
var r = assetFlt.LogReturn()[0];
mom = sc * r + mom * (1 - sc);
```

The main idea behind using the trading range instead of volatility is to put more emphasis on the price trend and filter out the daily fluctuations.







We used the same simple SPX strategy as before. The returns and the recovery time from drawdowns are the best we have achieved so far. While we are unsure if we like the very fast-paced adjustments to the filter period, the resulting momentum signal looks very promising.

Discussion & Next Steps

When we coded the ETF Rotation for TuringTrader, we were planning to conduct the following experiments:

- Isolate the market regime filter and implement separate strategies for bull and bear markets
- Potentially use the previously developed bond strategy for bear markets
- Experiment with more complex ranking schemes
- Experiment with dynamic position sizing
- Streamline the strategy's universe

However, we quickly realized that the internal momentum signal was the top issue to address. Without substantial noise reduction of this signal, all attempts to rank the assets by momentum are doomed to fail, rendering additional experiments meaningless.

We made a lot of new-to-the-world progress for the momentum signal. By now, it seems we have reached the limits of what we can achieve by adjusting filter periods based on internal indicators. The resulting momentum signal looks much cleaner than that of the conventional approach of measuring momentum over a fixed. This is especially remarkable, as the new momentum indicator has the ability to react much faster.

Further, the encouraging results of the simplistic momentum-based trend-following strategy support the belief that the new momentum indicator can potentially bring substantial improvements to asset-rotation strategies based on momentum-ranking.

We did have success with such strategies before, namely our Round-Robin strategy. However, the mechanism we used for Round Robin was very complex: It ran a swarm of about 120 instances of momentum indicators in parallel, selecting the best fit from these on a regular schedule to implement a walk-forward optimization. This complexity puts a burden on development that we would like to avoid, as it creates hard-to-meet requirements for the trading platform used. While TuringTrader is powerful enough to implement this type of strategies, most other platforms are not.

We recommend the following next steps:

- Code a custom indicator for the momentum-measuring method discussed above. This is a requirement to keep the code base clean and manageable.
- Update the existing strategy to use this new indicator.
- Experiment with the strategy, analyze the results, and make incremental improvements.

Because the new momentum signal is much faster and does not seem to depend on eliminating common-mode noise, we might be able to build a strategy without using a market-regime filter. In our opinion this would be a good thing, as it would allow more seamless transitions than a hard switch between bull and bear markets. We will likely still want to experiment with dynamic position sizing, and thinning out the universe.