

# L'indicateur RocketRSI

Une présentation de l'article "*RocketRSI - A Solid Propellant For Your Rocket Science Trading*" par John F. Ehlers

Présenté par Manuel Bolduc  
dans le cadre du cours GES816

# Sommaire de la présentation

- Contexte de l'article
  - Présentation de l'auteur
  - L'indicateur RSI
- Points Clés de la méthode Rocket RSI
- Implémentation python de RocketRSI
- Application de la méthode RocketRSI sur le cours de l'action RY
- Comparaison entre la méthode RocketRSI et le RSI

# Quelques mots sur l'auteur de l'article

John F Ehlers

- Auteur de plusieurs livres sur des stratégies d'investissement, dont *Rocket Science for Traders: Digital Signal Processing Applications*
- Nous avons notamment parlé en classe d'une de ses stratégies d'investissement, le *Zero-Lag EMA*



But de l'article: la méthode RocketRSI

La méthode RocketRSI est conçue afin de rendre l'utilisation de l'indicateur RSI **plus flexible** et **plus facile à interpréter statistiquement**

# L'indicateur RSI (Indice de force relative)

$$RSI = 100 * (1 - 1 / (1 + H/B))$$

Indicateur RSI sur le cours de l'indice RY

**H**: moyenne des fermetures  
montantes des n derniers jours

**B**: moyenne des fermetures  
baissières des n derniers jours

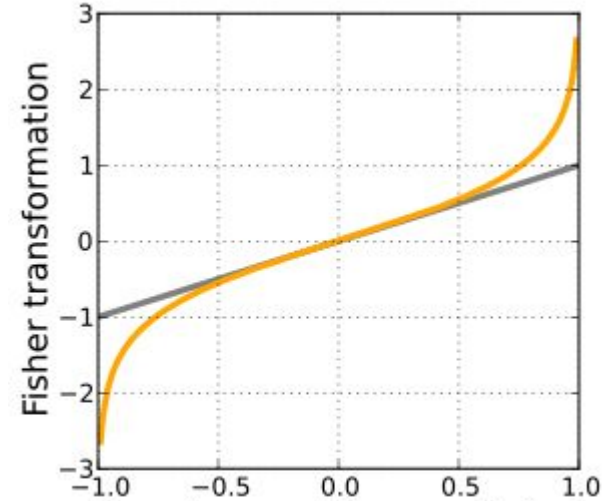
**RSI < 30** : l'action est  
survendue

**RSI > 70**: l'action est  
surachetée



# Points clés de la méthode RocketRSI

- Simplifier le calcul de l'indicateur RSI afin qu'il oscille entre -1 et +1 plutôt que 0 et 100
- Ajouter une étape de lissage de données avant le calcul de l'indicateur RSI pour avoir un signal plus clair - utilisation du **SuperSmoother**
- Application d'une transformée de Fisher aux données afin de dégager des signaux statistiques clairs



Transformée de Fisher. Tirée de Wikipedia

# Implémentation de la méthode RocketRSI en Python

```
def SuperSmootherFilter(prices, smoothlength = 8):  
  
    Filt = np.zeros_like(prices)  
  
    a1 = np.exp(-np.sqrt(2)*np.pi/smoothlength)  
    b1 = 2*a1*np.cos(np.sqrt(2)*np.pi/smoothlength)  
    c2 = b1  
    c3 = -a1*a1  
    c1 = 1 - c2 - c3  
  
    for i in range(len(prices)):  
        if i < 3:  
            Filt[i] = prices[i]  
        else:  
            Filt[i] = c1*(prices[i] + prices[i-1])/2  
            Filt[i] += c2*Filt[i-1] + c3*Filt[i-2]  
    return Filt
```



Le SuperSmootherFilter, pour lisser les données d'entrées

```
def myrsi(prices, period=10):  
  
    prices = np.array(prices)  
    myrsi = np.zeros_like(prices)  
  
    deltas = np.diff(prices)  
    prices_up = deltas*(np.sign(deltas) + 1)/2  
    prices_down = -deltas*(1 - np.sign(deltas))/2  
  
    for i in range(len(prices)):  
        if i < period - 1:  
            myrsi[i] = np.nan  
        else:  
            CU = np.sum(prices_up[i - period+1:i+1])  
            CD = np.sum(prices_down[i - period+1:i+1])  
            myrsi[i] = (CU - CD)/(CU + CD)  
  
    return myrsi
```



L'équation adapté de RSI pour que l'indicateur oscille entre -1 et 1

# Implémentation de la méthode RocketRSI en Python (suite)

```
prices = np.array(daily_ry['Close'])
mom_prices = momentum(prices)
smooth_prices = SuperSmootherFilter(mom_prices)
myrsi_prices = myrsi(smooth_prices)

for i in range(len(myrsi_prices)):

    if myrsi_prices[i] > 0.999:
        myrsi_prices[i] = 0.999

    if myrsi_prices[i] < -0.999:
        myrsi_prices[i] = -0.999

rocketrsi = 1/2*np.log((1+myrsi_prices)/(1-myrsi_prices))
```



L'algorithme pour calculer le RocketRSI, avec les fonctions définies auparavant

```
def momentum(prices, period = 10):

    prices = np.array(prices)
    momentum = np.zeros_like(prices)

    for i in range(len(prices)):

        if i < period - 1:
            momentum[i] = 0.

        else:
            momentum[i] = prices[i] - prices[i - period]

    return momentum
```



La fonction de momentum utilisée dans le calcul du RocketRSI



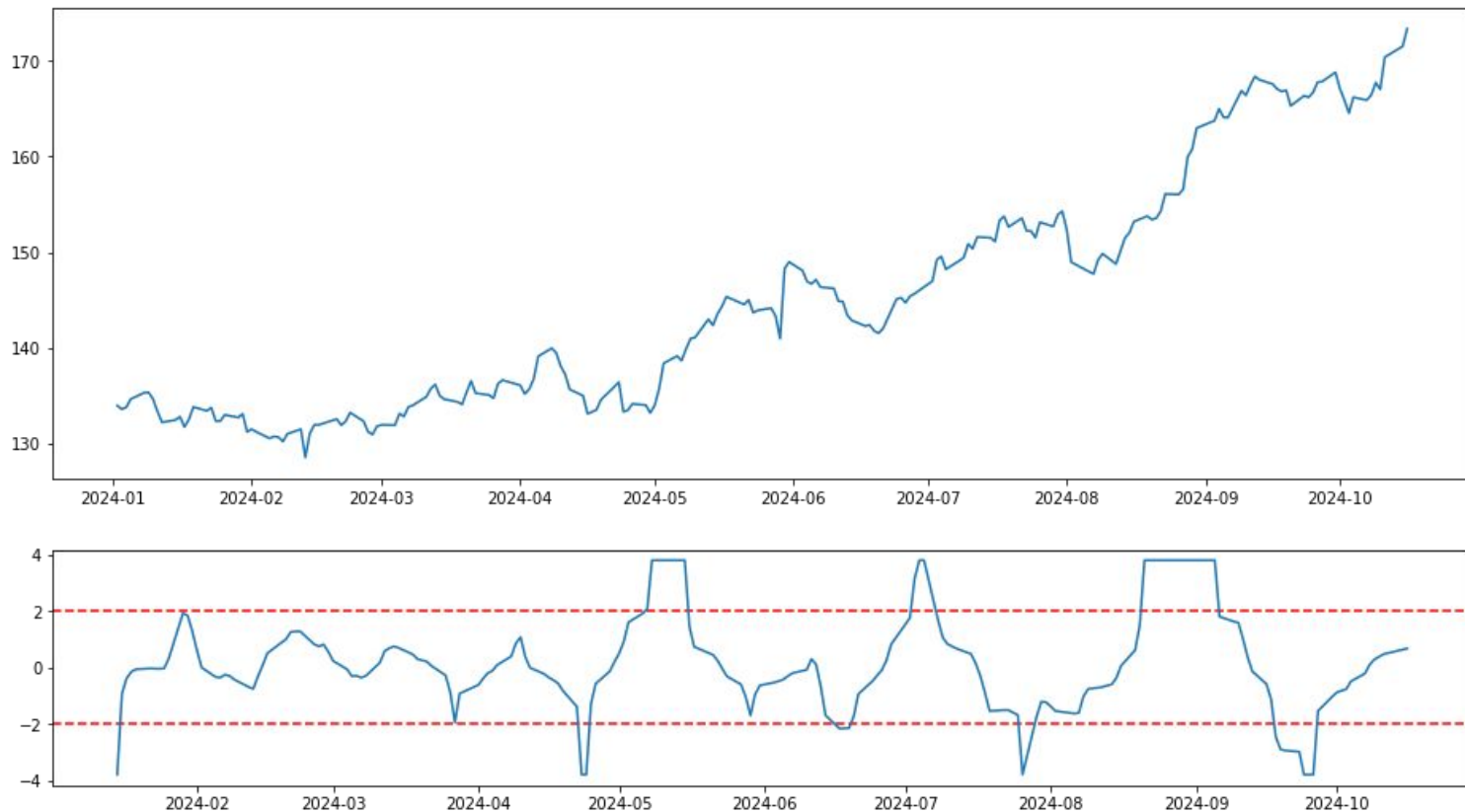
# Résultats sur le cours de l'action RY (banque royale du Canada)

**RocketRSI > 2:**

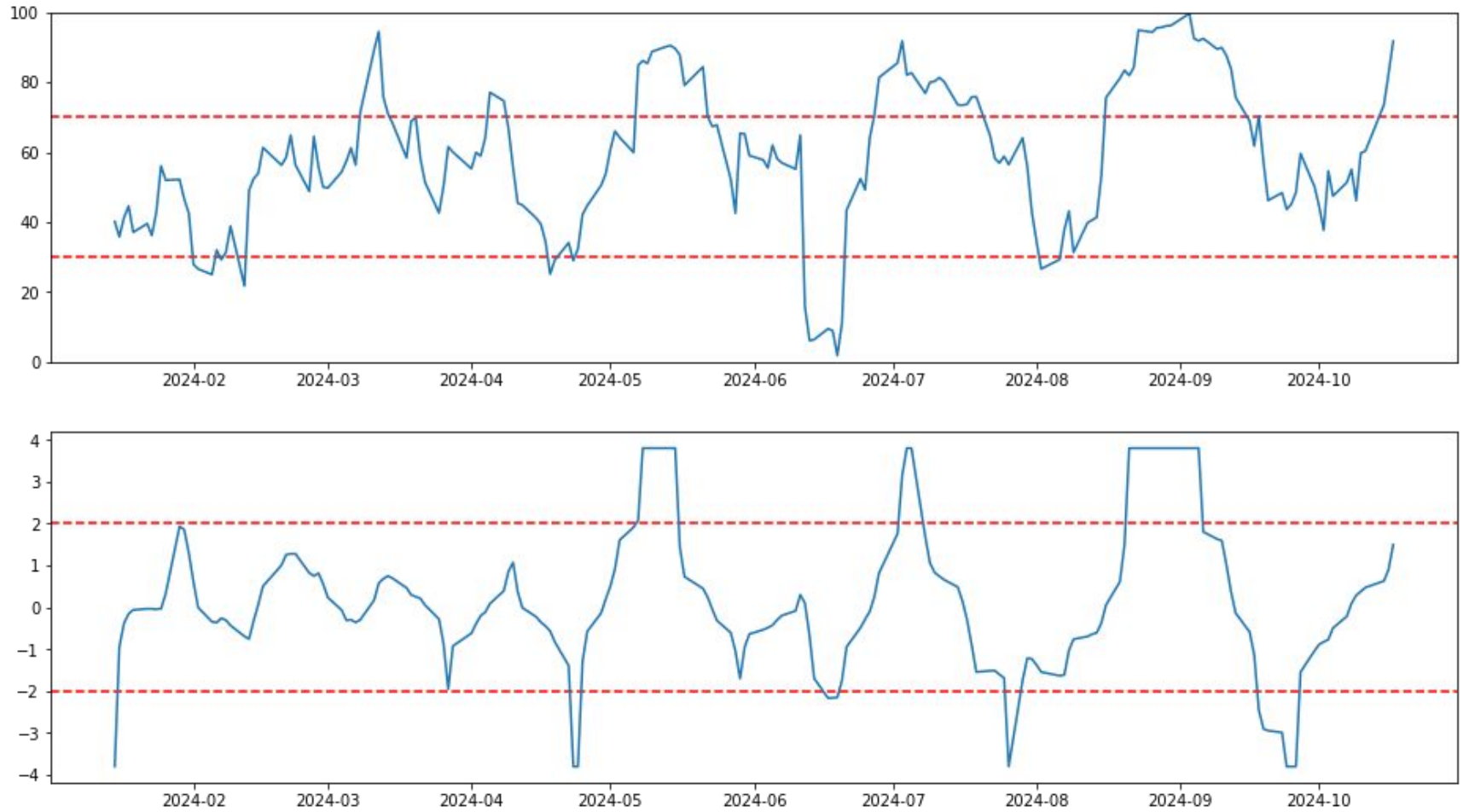
L'action est  
surachetée

**RocketRSI < -2:**

L'action est  
survendue



# Comparaison des signaux générés par le RSI (en haut) et le RocketRSI (en bas)



# Bibliographie

“Fisher Transformation.” *Wikipedia*, 2 June 2024. *Wikipedia*, [https://en.wikipedia.org/w/index.php?title=Fisher\\_transformation&oldid=1226838743](https://en.wikipedia.org/w/index.php?title=Fisher_transformation&oldid=1226838743). Consulté le 14 Octobre

Ehlers, John F. *Rocket Science for Traders: Digital Signal Processing Applications*. Wiley, 2001.

Ehlers, John F. “RocketRSI - A Solid Propellant For Your Rocket Science Trading.” *Stocks & Commodities*, vol. 36, no. 05, 2018, pp. 8–12.

Miresco, Edmond. *Notes de Cours - GES816*. École de Technologie Supérieure.

NOTE: le code présenté dans la présentation est disponible au lien suivant:

<https://github.com/bolducmanuel/adaptivemovingaverage/blob/main/RocketRSI.ipynb>