Pseudo-code RSM

Agent checks identity threat {  
 compares Threshold towards Ratio:(opposing opinions weight ^ 2 / supporting opinions weight)  
}

If agent threaten {  
 choose agent X or medium X or yourself with probability proportional to her/its weight  
 if agent X is exhausted:(Energy <= 0) {  
 chose new agent X or medium X or yourself  
 }  
 pick threatening piece of info TPI proportionally to its weight from your memory  
 send TPI to agent X {  
 agent X stores TPI in her own memory with weight of myself  
 if TPI threatens agent X {  
 agent X lowers the weight of myself/calling agent  
 }  
 if TPI not threatens agent X {  
 agent X raises the weight of myself/calling agent  
 }  
 }  
 ask agent/medium X for opinion {  
 agent/medium X draws opinion according attitude Att, variance Sigma and skewness Gamma  
 }  
 store opinion of agent X in memory with weight of agent X  
 if communicating with myself {  
 chose opinion from memory proportionally to its weight  
 store memorized opinion again as new opinion  
 store weight of opinion according actual weight of original agent  
 store X as agent of original opinion  
 }  
 check identity threat again  
 if opinion X helps lower identity threat {  
 update weight of agent X to higher value  
 }  
 if opinion X doesn’t help lower identity threat {  
 update weight of agent X to lower value  
 }  
 lower your Energy daily reservoir by Epsilon  
 If exhausted:(Energy <= 0) or not threaten:(Threshold > Ratio) {  
 stop the cycle  
 }  
 If not exhausted (Energy > 0) and threaten:(Threshold < Ratio) {  
 do the cycle once more again (from ‘chose agent X or …’)  
 }  
}

Wait until the end of day – other agents might ask your opinion (described above), but if you are exhausted, there will be nothing until end of the day

Agent updates Attitude (see next page)

Agent updates Memory {lower weights of opinions in the memory by Ebbinghouse curve}

Agent updates:(refresh Energy to full reservoir).

Updating Attitude pseudo-code:

Individual:

New attitude = Mean(Attitude, mean(opinions in memory weighted)) \* 1 / Ratio:(if Ratio > 1, take Ratio = 1)

OR

If Attitude > mean(opinions in memory weighted) {  
 Attitude = Attitude – 1 / Ratio  
}

If Attitude < mean(opinions in memory weighted) {  
 Attitude = Attitude + 1 / Ratio  
}

Social:

Attitude = Attitude ± (abs(Attitude - mean(attitudes of agents in radius Sigma))/2 \* 1 / Ratio)

OR

Attitude = (Attitude + mean(attitudes of agents in radius Sigma))/2

Updating Sigma pseudo-code:

Individual:

Threat -> lower Sigma (radius is narrower)  
No threat -> raise Sigma (radius is wider)

Social:

Sigma of agent = mean(Sigma of agents inside Sigma of agent)