

Emp: General Purpose Empirical Management Procedure

A customizable MP that uses relative abundance indices to calculate TAC advice using a harvest control rule.

Tom Carruthers (tom@bluematterscience.com)

2025-10-03

Contents

Empirical Management Procedures	1
Function Arguments	2
Generic arguments x, Data, reps	2
Inds: which indices to use in calculating TAC advice?	2
I_freq: how frequently will indices be available?	3
I_wt: how should indices be weighted in the calculation of management advice?	3
calib_yrs: how many recent historical years of index and catch data are to be used in calibrating the harvest control rule?	3
enp_mult: degree of smoothing of index data	4
Ind_fac & curI_2_target: mapping harvest control rule axes to empirical data.	4
HCR_CP_B & HCR_CP_TAC: control points of the hockey stick harvest control rule	5
TACrng: The minimum and maximum allowable catches	6
delta_down: Minimum and maximum rates of downward TAC adjustment	6
delta_up: Minimum and maximum rate of upward TAC adjustment	6
resp: TAC change responsiveness	6
Mode: index ratio versus index target MP	7
Default Specification	7
Example code	7
Equations	8
Emp code	8
References	10

Empirical Management Procedures

Empirical management procedures (MPs) calculate management advice from data inputs without fitting statistical fisheries models to data. This distinction between empirical and ‘model-based’ MPs is less clearly delineated than it might at first seem since many empirical MPs fit smoothers to data or impose simplistic fishery assumptions in their calculations. Nonetheless in most settings where they have been adopted, empirical MPs are computationally efficient, robust, do not require subjective supervision and are reproducible.

Model-based MPs can involve model fitting diagnostics that might compromise TAC advice in future years. Additionally, model-based MPs blur the lines between stock assessment and MSE which can lead to confusion

among managers and stakeholders. For these various reasons, empirical MPs are often seen as a primary goal of MSE: simple, transparent rules for rapid and reproducible management advice of demonstrated robustness. An empirical MP aiming for approximately constant exploitation rate was the end product of the Atlantic Bluefin Tuna MSE process (Butterworth and Rademeyer 2024). Similarly an index-based empirical MP was also adopted for North Atlantic swordfish (ICCAT 2024).

A key limitation of empirical MPs is that they may have insufficient flexibility to adapt to strongly varying operating model conditions. For example, consider two operating models that fit historical data comparably but are an order of magnitude different in scale. In the small stock scenario, an empirical MP can make adjustments to TACs to compensate for declining indices, however the model-based MP that estimates carrying capacity may adapt more reliably to any apparent scale information, for example aiming more consistently for an appropriate productive stock size.

Function Arguments

OpenMSE includes a general-purpose empirical management procedure that uses relative abundance data to set TACs. The MP can be customized to include multiple indices, weight those indices, modify catch advice according to a harvest control rule, simulate data-lags across varying indices, smooth data to varying degrees and control overall rates of TAC change.

Emp is not necessarily intended to be used in management advice but rather for teaching purposes and as an exploratory tool for custom empirical MP development.

To learn more about Emp you can use the inline R help:

```
library(openMSE)
?Emp
```

You can also just take a look at the raw code for the MP function:

```
Emp
```

Generic arguments x, Data, reps

All openMSE MPs have the arguments x (simulation number), Data (simulated data) and reps (number of samples of advice for stochastic MPs).

Inds: which indices to use in calculating TAC advice?

The first key argument of Emp is Inds which is a vector of positive integers. Each integer corresponds with a relative abundance index that is being simulated. To better understand this argument we can take a look at an example dataset:

```
avail('Data')           # OpenMSE objects of class 'Data' in the workspace
Data = SimulatedData    # Make a copy of SimualtedData
slotNames(Data)          # The various slots of an object of class 'Data'
class?Data               # In line help on Data objects
dim(Data@AddInd)         # A 3D array [simulation, index, year]
```

You can see that the AddInd slot of the Data object has three dimensions: simulation, index and year. The Inds argument selects which of the indices will be used. So a vector c(2, 4, 10) will use three indices, those in positions 2, 4 and 10 of the AddInd array.

If you were to specify NA for the Inds argument, the MP defaults to a generic relative abundance index that is simulated and assumed to reflect total biomass (Data@Ind).

I_freq: how frequently will indices be available?

The I_freq argument allows you to control the frequency of index availability. In many settings indices are collected and processed at different temporal resolutions. For example, CPUE indices available on an annual basis and surveys every 3 years.

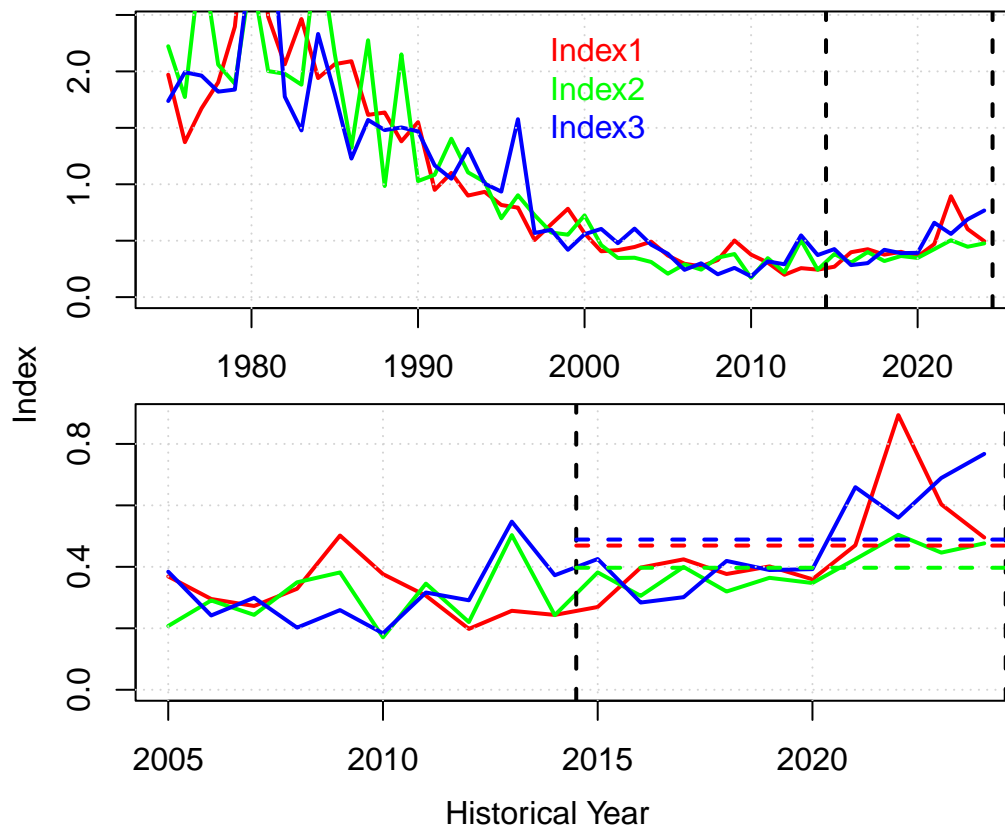
I_freq allows you specify this, allowing you to maintain realism in data availability or simulation test the cost / benefit of data collection frequency. I_freq is a vector of the same length at Inds. For example, given the Inds example above, an I_freq vector of c(1,3,1) would mean index 2 is provided every year, index 4 is provided every three years, index 10 is available every year.

I_wt: how should indices be weighted in the calculation of management advice?

Its possible to adjust the relative contribution of the various indices in the calculation of current stock level. Again, this is a vector of positive real numbers as long as the Inds argument. A zero value weight ignores the index.

calib_yrs: how many recent historical years of index and catch data are to be used in calibrating the harvest control rule?

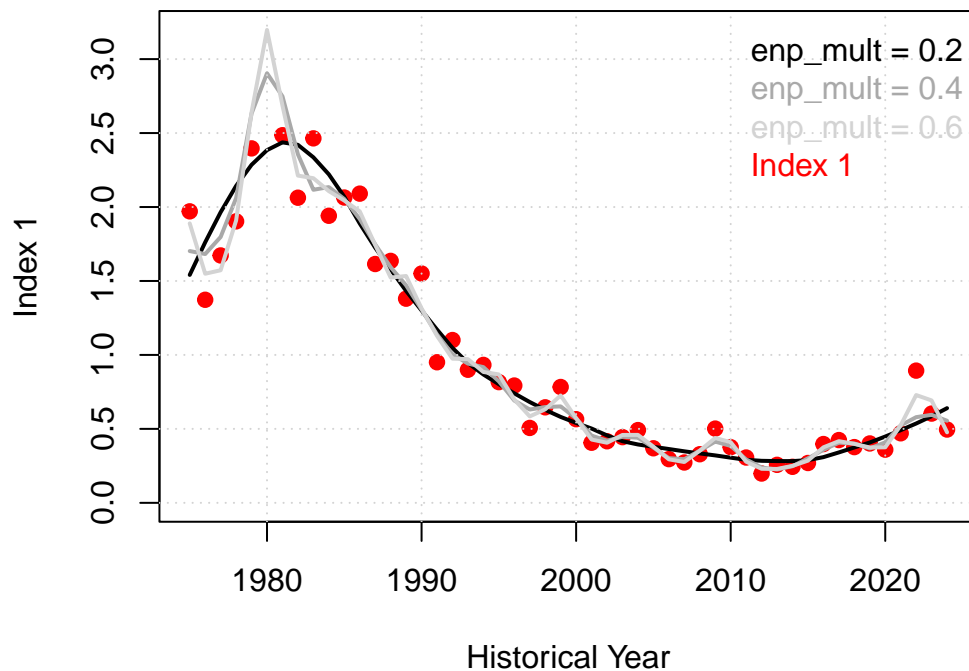
Future TAC advice will be calculated according to index levels relative to a target level. To set the target level for the indices we first specify a reference level based on the historical data. This is determined by the calibration years argument that takes the mean of each index over the the most recent calib_yrs number of years. In the plot below, the most recent 10 years of each index is used to define a reference level that is the mean over those last 10 historical years (dashed colored lines).



enp_mult: degree of smoothing of index data

In order to filter indices to better characterize level and trend, a loess smoother is applied to the index data. This is to prevent TAC advice from fluctuating due to observation error rather than biomass levels. The effective number of parameters of that smoother determines how well it fits the data. When ENP is high, the smoother fits the data better but removes less noise from observation error.

In order to keep the degree of smoothing constant as the index time series increases (in future simulation years) this is parameterized as an ENP multiplier, where the effective number of parameters is a fixed fraction of the length of the index time series.

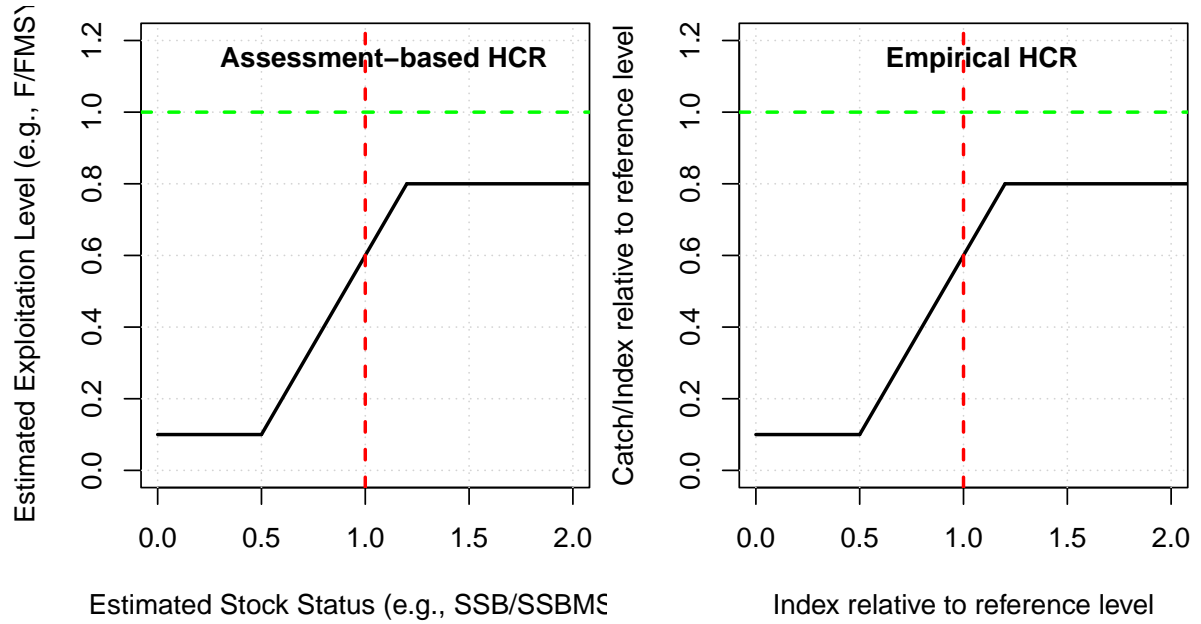


The level of smoothing should reflect the degree of expected variation in the vulnerable biomass inferred by the index. In the example above, inferred biomass trends are apparently fluctuating on a relatively long time scale that is well approximated by a lower value of `enp_mult`.

Ind_fac & curI_2_target: mapping harvest control rule axes to empirical data.

In a conventional hockey-stick harvest control rule from which management advice is calculated from stock assessment model outputs, the x and y axis are typically estimated quantities such as stock status (SSB/SSBMSY) and Exploitation rate (F/ FMSY).

In an empirical MP we wish to remap this according to our empirical data, not assessment-model based estimates:



This is parameterized in Emp according to the historical period defined by `calib_yrs`. Over this recent historical time period a mean level of each index was calculated. The argument `curI_2_target` defines where that level is relative to the reference level. For example, if you wish to specify an HCR that assumes that index levels over the last 10 years corresponded to half of SSBMSY levels, `calib_yrs` would be set to 10 and `curI_2_target` would be set to 0.5. From now on, any index data used by the HCR of the MP will be scaled this way. In this case an index level three times the historical reference level will be interpreted in the HCR as $SSB/SSBMSY = 1.5$.

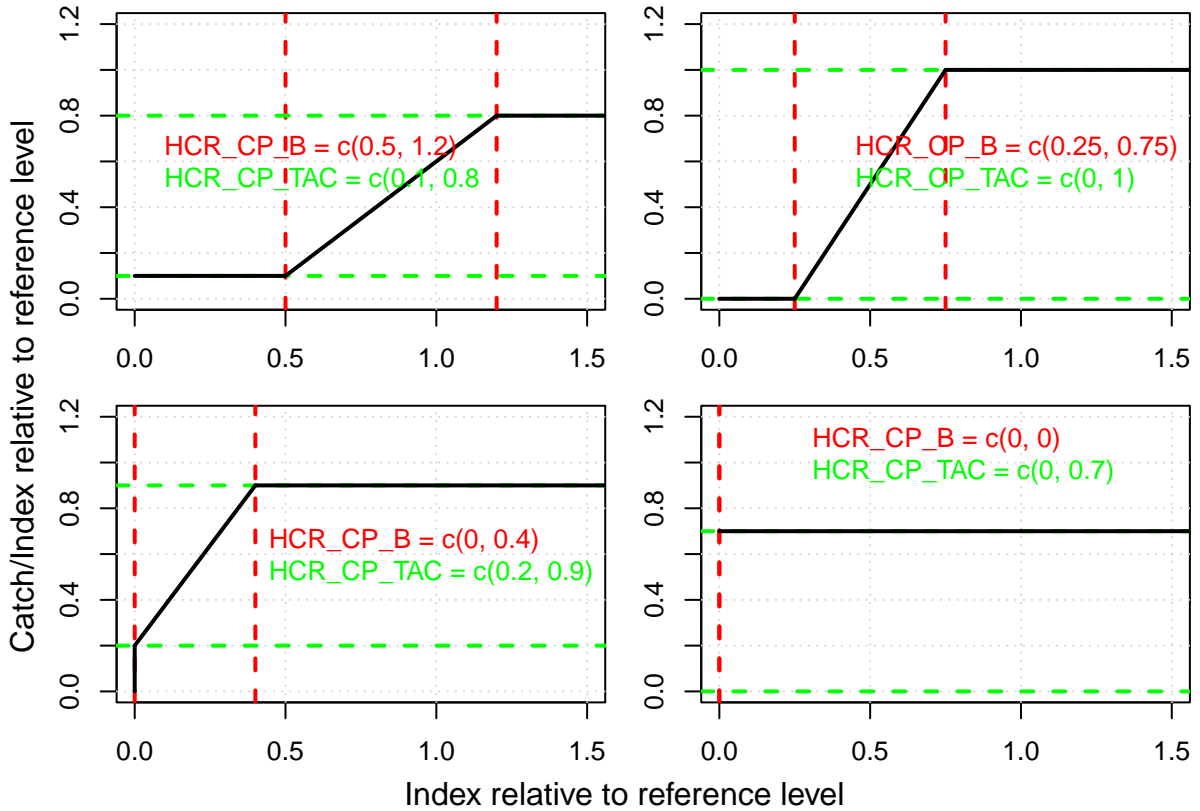
The y-axis of the HCR is defined in terms of catch per index over the historical reference period defined by `calib_yrs`. For example we may wish the empirical HCR to assume that over the last 10 years the stock was overfished by 100% (F/FMSY was 2). In this case the catch per index is twice the target level and the `Ind_fac` parameter would be 0.5. The reference exploitation level (implied FMSY) is half the most recent catch per index level.

It follows that the `Ind_fac` and `curI_2_target` parameters alter the degree to which the empirical MP may overfish and maintain underfished status.

HCR_CP_B & HCR_CP_TAC: control points of the hockey stick harvest control rule

The `Ind_fac` and `curI_2_target` parameters have allowed us to interpret new index data in terms of the two axes of a hockey stick HCR. Now we have to specify the control points that determine the inflection points of the HCR. These are two sets of coordinates specified according to inferred biomass on the x axis (HCR_CP_B) and inferred exploitation rate on the y-axis (HCR_CP_TAC). These coordinates are fractions of the reference level for the index and catch per index.

The best way to explain these arguments to Emp is to plot some examples:



TACrng: The minimum and maximum allowable catches

The TACrng argument simply allows the user to put hard limits on the range of TAC advice that can be recommended by the MP.

delta_down: Minimum and maximum rates of downward TAC adjustment

In some cases small reductions in TAC may not be worth a revision to an existing TAC. In such cases setting a minimum rate of decrease (say 2%) might save administrative overhead.

In order to provide less variable TAC advice it may be desirable to constrain TAC changes to a maximum rate of downward adjustment, for example 10%.

In this example the delta_down argument would be set to $c(0.02, 0.1)$

delta_up: Minimum and maximum rate of upward TAC adjustment

Similarly to delta_down a user might want to limit upward TAC adjustments to only those larger than 1 percent and allow for up to 5 percent increases. In this example the delta_up argument would be set to $c(0.01, 0.05)$.

resp: TAC change responsiveness

Rather than use the delta parameters to set hard limits on TAC adjustments, you may wish to simply make the TAC change less responsive. Let us define the TAC adjustment: $TAC_{adj} = \text{proposedTAC} / \text{oldTAC}$. The resp parameter works by reducing the adjustment ratio in logspace to: $\text{newTAC} = \text{oldTAC} \times \exp(\log(TAC_{adj}) \times \text{resp})$.

It follows that resp values of less than 1 dampen the rate of TAC change. In many MSE cases it may be possible to use the resp parameter to obtain substantial gains in catch stability with limited impacts on yield and conservation performance.

Mode: index ratio versus index target MP

By default Emp is specified to set TACs according to a desired exploitation rate (Mode 1, which uses the HCR above). However you can also specify Emp to aim for a target index level by setting Mode to 2. In this case TAC adjustments are made to maintain a level of the index specified by curI_2_target. The TACrng, delta_up, delta_down and resp parameters still operate on the TAC advice in index target mode (Mode = 2)

Default Specification

At default argument values the Emp function:

- (Inds = NA) Uses the single biomass-tracking index in the Data object Data@Ind rather than the additional indices slot Data@AddInd. The properties of that are determined by OM@Iobs (the coefficient of variation, lognormal) and OM@beta (the hyperstability parameter)
- (I_freq = NA) Assumes all indices are sampled every year
- (I_wt = NA) Equally weights all indices
- (calib_yrs = 2) Calibrates reference yield and indices to the last two historical years
- (enp_mult = 0.2) Smoothes indices using a loess smoother that has 0.2 * ny effective number of parameters.
- (Ind_fac = NA) Calculates true simulated FMSY relative to F over the last calib_yrs historical years and sets the target to 75 of this level (aiming for 0.75% FMSY)
- (curI_2_target = NA) Calculates the true simulated vulnerable biomass over calib_yrs relative to true VBMSY and sets the target to 125% of this level (aiming for 125% BMSY)
- (HCR_CP_B = c(0, 0)) A constant exploitation policy
- (HCR_CP_TAC) = c(0, 1) Fishes at zero below the lower biomass control point and at the exploitation rate target above the upper biomass control point.
- (TACrng = NA) TACs are between zero and 100 times the maximum historical catch (no TAC limit)
- (delta_down = c(0.01, 0.5)) Up to 50% decreases in TAC. Changes of less than 1% are ignored.
- (delta_up = c(0.01, 0.5)) Up to 50% increases in TAC. Changes of less than 1% are ignored.
- (resp = 1) TAC changes are not adjusted for responsiveness
- (Mode = 1) TACs are set according to a harvest control rule that imposes exploitation rate changes in response to index data.

Example code

```
library(openMSE)
```

```
Emp_CER = Emp # Emp defaults to constant exploitation rate target Exp Rate = 0.75 FMSY, target B = 1.25
formals(Emp_CER)$HCR_CP_TAC = c(0, 4/5) # since the set up is 75% FMSY this is equivalent to 60% FMSY
```

```
Emp_5_10 = Emp
formals(Emp_5_10)$HCR_CP_B = c(0.5, 1)*(1/1.25) # Exp Rate ramps from 0.5 to 1 BMSY
```

```

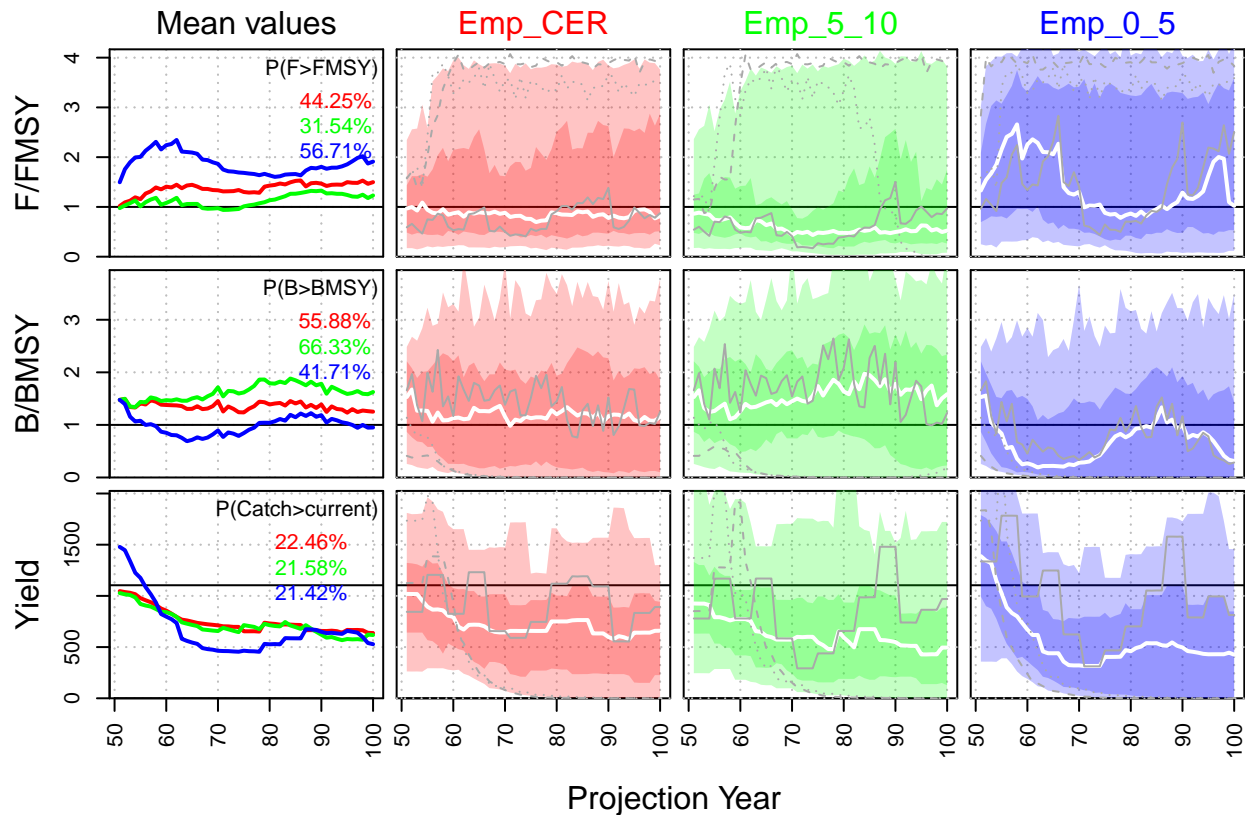
formals(Emp_5_10)$HCR_CP_TAC = c(0,1/0.75)          # target Exp Rate = FMSY

Emp_0_5 = Emp_5_10
formals(Emp_0_5)$HCR_CP_B = c(0, 0.5)*(1/1.25)      # Exp Rate ramps from 0 to 0.5 BMSY

class(Emp_CER) = class(Emp_5_10) = class(Emp_0_5) = 'MP'

testOM@nsim = 48
demoMSE = runMSE(testOM, c("Emp_CER", "Emp_5_10", "Emp_0_5"))
Splot(demoMSE)

```



Equations

The Emp function follows a series of steps that go from index and catch calibration to index processing, the harvest control rule and finally the adjustment of a proposed TAC to meet various TAC change constraints.

Emp code

Emp is intended to be a teaching tool but also as a starting point for making user-designed empirical MPs. The code is available from the command prompt and is reasonably straightforward to follow (takes the same steps as the equations above).

```

Emp = function (x, Data, reps = 1, Inds = NA, I_freq = NA, I_wt = NA, calib_yrs = 2, enp_mult = 0.3,
  Ind_fac = NA, TACrng = NA, delta_down = c(0.01, 0.5), delta_up = c(0.01, 0.5),
  resp = 1, curI_2_target = NA, HCR_CP_B = c(0, 0), HCR_CP_TAC = c(0, 1), Mode = 1){

```

```

MPrec = Data@MPrec[x]
ny = length(Data@Cat[x, ])
ystart <- which(!is.na(Data@Cat[x, ]))[1]
yind <- ystart:ny
LHYr = Data@LHYear
LHYrInd = match(LHYr, Data@Year)
CurYr = max(Data@Year)
Year <- Data@Year[yind]
C_hist <- Data@Cat[x, yind]

if (is.na(Inds[1])) {
  I_hist <- array(Data@Ind[x, yind], c(1, ny))
} else {
  I_hist <- array(Data@AddInd[x, Inds, yind], c(length(Inds), ny))
}

nI = nrow(I_hist)

if (is.na(I_freq[1])) I_freq = rep(1, nI)

if (is.na(I_wt[1])) I_wt = rep(1, nI)

if (is.na(Ind_fac)) {
  FMSY = Data@Misc$ReferencePoints$ByYear$FMSY[, LHYrInd]
  FM = Data@Misc$FleetPars$Find[, LHYrInd] * Data@Misc$FleetPars$qs
  Ind_fac = mean(FMSY/FM) * 0.75
}

if (is.na(curI_2_target)) {
  Dep = Data@Misc$StockPars$Depletion
  BreI = Data@Misc$StockPars$BMSY_B0
  curI_2_target = mean(Dep)/mean(BreI * 1.25)
}

if (is.na(TACrng)) TACrng = c(0, 100 * max(Data@Cat))

I_keep = MSEtool::doIfreq(I_hist, I_freq, LHYr, CurYr, Year)
caliby = LHYrInd - (calib_yrs - 1):0
calibmuI = apply(I_keep[, caliby, drop = F], 1, mean, na.rm = T)
ref = mean(calibmuI/curI_2_target, na.rm = T)
calibmuC = mean(C_hist[caliby], na.rm = T)
C_I = calibmuC/calibmuI
nkeep = sum(I_freq != 0)
I_smth = array(NA, c(nkeep, ny))

for (j in 1:nkeep) I_smth[j, ] = MSEtool::smoother(I_keep[j, ], enp_mult = enp_mult)

est = weighted.mean(I_smth[, ny], I_wt, na.rm = T)

if (Mode == 1) {
  TACbyI = I_smth[, ny] * C_I
  TACtemp = mean(TACbyI, na.rm = T) * Ind_fac
  if (is.na(TACtemp) | is.null(TACtemp)) TACtemp = Data@MPrec[x]
}

```

```

    trial_TAC = MSEtool::TACfilter(TACtemp)
    HCRadj_TAC = MSEtool::doHCR(trial_TAC, est = est, ref = ref, CP = HCR_CP_B, CPy = HCR_CP_TAC)
    mod = exp(log(HCRadj_TAC/MPrec) * resp)
  } else {
    mod = exp(log(est/ref) * resp)
  }

  MSEtool::doRec(MPrec, mod, delta_down, delta_up, TACrng)
}

```

References

- Butterworth, D.S. & Rademeyer, R. 2022. BR CMP AS AT JUNE 2022. Col. Vol. Sci. Pap. ICCAT. 79(3): 587-597
- ICCAT 2024. North Atlantic Swordfish MSE: Final Results & Decision Guide. PA4_812/2024