Harvest Control Rules

MSE with FLR/a4a course. JRC (Ispra) 25-29 November 2019

Iago MOSQUEIRA

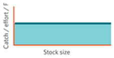
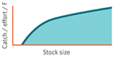
Wageningen Marine Research (WMR), IJmuiden, The Netherlands.

WAGENINGEN
UNIVERSITY & RESEARCH

100years
1918 — 2018

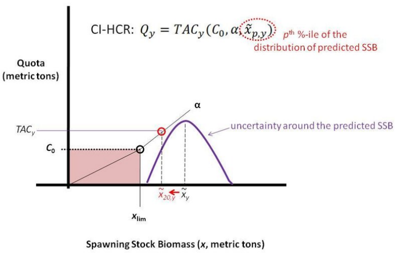
Harvest Control (Decision) Rules

Figure 1
Main Types of Harvest Control Rules

HCR type	Description	What it looks like
Constant	Allows for a constant level of fishing based on one value, regardless of stock status. The single value could be mortality (F), total allowable catch, days at sea, etc.	
Threshold	Fishing is allowed at a single target level until a limit is reached, at which point fishing is stopped.	
Step	Incorporates steps so higher fishing levels are permitted as the stock's status improves.	
Sliding (simple linear)	A sliding rule allows for a continuous adjustment in fishing controls. Higher fishing levels are permitted with improved stock status.	
Sliding (complex linear)	Same as above, but linear combinations can be complex, meaning that different responses may be triggered at different thresholds.	
Sliding (nonlinear)	Similar to the sliding forms, but the adjustments are nonlinear. This may be logarithmic (i.e., a smooth increase in fishing levels as stock status improves, as shown) or logistic (more S-shaped—i.e., a smooth increase up to a constant control measure at larger stock sizes).	

Source: Aaron M. Berger et al., *Introduction to Harvest Control Rules for WCPO Tuna Fisheries*

The Confidence Interval HCR Implementation



Types of HCR

By input

- Empirical
- Model-based

By output

- F
- Catch
- Harvest rate
- Effort

Model-based HCRs

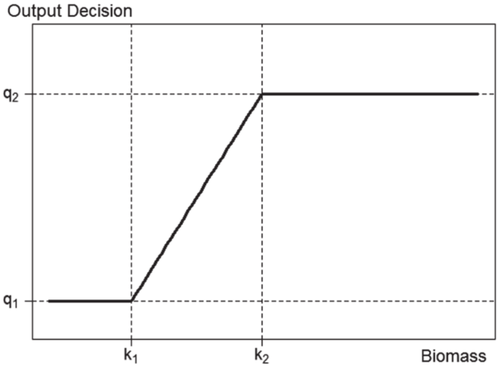
- Statistically combine observations.
- Incorporate observation and process error.
- Extract signal from noise.

But

- Subjective weighting of datasets.
- Sensitivity runs and diagnostics.

A standard 40/10 HCR

$$D = \begin{cases} q_1 & B \leq k_1 \\ q_1 + (B - k_1)\dfrac{q_2 - q_1}{k_2 - k_1} & k_1 < B < k_2 \\ q_2 & B \geq k_2 \end{cases}$$



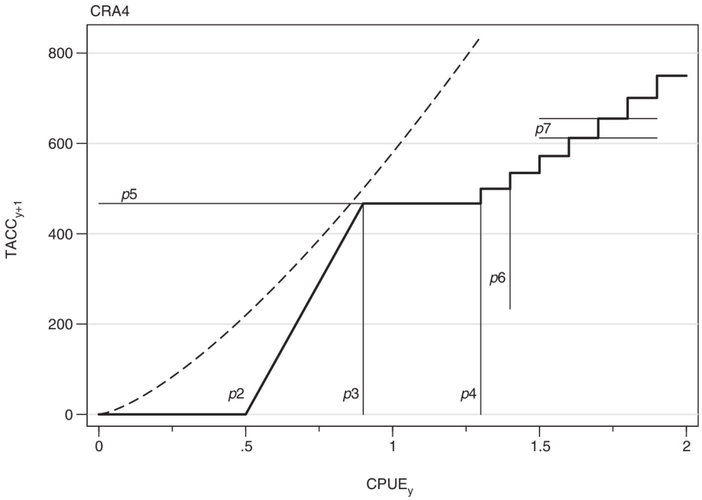
Empirical HCRs

- Simpler to explain.
- Usable for many more stocks.

But

- Noise and bias have greater effect.
- Missing data.

A CPUE (kg/pot) HCR



Mixed HCRs

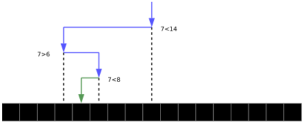
- Simple models: biologically plausible filters.
- Efficienty to run.
- Look for signal.

CCSBT HCR

- Empirical HCR on random walk model of CPUE
- CPUE (LL) + Arial survey / close kin analysis

Tuning for HCR parameters

- Find values of HCR parameters that return the desired management objective
- Choose the most responsive parameter
- Bisection (binary) search from upper/lower values



- Compute performance indicator and chose side

moving.phcr

```
movingF.phcr <- function(stk, frp="f0.1", model="missing",
  interval, args, hcrpars, tracking) {
    ay <- args$ay
    iy <- args$iy
    if(ay==iy | (ay-iy)%%interval==0){
        if(!missing(model)){
            sr0 <- fmle(as.FLSR(stk, model=model))
            hcrpars <- c(refpts(brp(FLBRP(stk, sr0)))[frp,"harves
        } else {
            hcrpars <- c(refpts(brp(FLBRP(stk)))[frp,"harvest"])
        }
    }
    list(hcrpars=hcrpars, tracking=tracking)
}
```

moving.hcr

```
movingF.hcr <- function(stk, hcrpars, args, tracking){
    ay <- args$ay
    # rule
    if(!is(hcrpars, "FLQuant"))
    hcrpars <- FLQuant(hcrpars, dimnames=list(iter=dimnames(stk@c

  # create control file
    ctrl <- getCtrl(c(hcrpars), "f", ay+args$management_lag, dim(

  # return
    list(ctrl=ctrl, tracking=tracking)
}
```

ices.hcr

```
ices.hcr <- function(stk, fmin, ftrg, blim, bsafe, args, tracking
    # rule
    ssb <- ssb(stk)[, ac(ay-ssb_lag)]
    fout <- FLQuant(fmin,
    dimnames=list(iter=dimnames(ssb)$iter))
    fout[ssb >= bsafe] <- ftrg
    inbetween <- (ssb < bsafe) & (ssb > blim)
    gradient <- (ftrg - fmin) / (bsafe - blim)
    fout[inbetween] <- (ssb[inbetween]-blim)*gradient+fmin
    [...]
    list(ctrl=ctrl, tracking=tracking)
}
```