Navigation Problem of Speed research for parameters dependence

Daniel Paliura

3/11/2021

Introduction

This document created as additional to 'Control Theory' subject first laboratory work 'Navigation Problem of Speed'. In this document will be researched dependence of input parameters on problem solving. Also reasons of effect will be explained if it's possible.There wasn't said anything about boat turn speed so boat turn speed is neglected.

Formulation of the Navigation Problem of Speed

It's boat on water surface which has to catch the immovable target placed on the same water surface. Water has stream moving with some speed. Formalized problem: to find boat control that moves boat to reach target. Boat is defined as material point which moves with constant speed v relative to water surface. Initially boat is in origin - point $(0, 0)$ - of plane xOy, where target does not moves, which means that water surface moves relatively to plane. Stream vector is parallel to ordinate axis (Ox) in each point of plane and perpendicular to abscissa axis (Oy) if it isn't zero vector. Absolute value of stream vector depends on y coordinate and is described by expression:

$$s(y) = s_0 \cdot f(y),$$

where
s_0 - stream speed multiplier or initial stream speed,
$f(y)$ - some defined function of stream speed dependence. Target is in point distanced on l from origin and angle between x axis and vector from origin to target is ϕ, so target is in point $x^* = l \cdot cos(\phi), y^* = l \cdot sin(\phi)$.

Solution method provided

To solve this problem was spelled aiming method, which chooses best control vector in current point considering stream speed in point. In this method the speed of stream is considered as constant for small time interval $\tau = \frac{l}{vN}$, where N is some large number. I set $N = 1e3$.
According to analytic calculations (Lagrange polynomial optimization) method uses next recurrent formulae:

$$\lambda(t_k) = v\tau\sqrt{(x^* - x_k - s(y_k)\tau)^2 + (y^* - y_k)^2} - (v\tau)^2,$$

where
$t_k = k\tau$ - moment of time on k iteration,
x_k, y_k - coordinates of boat position in moment t_k,
$k = 0 \cdots K, N \leq K \in \mathbb{N}$
and $t_0 = 0, x_0 = 0, y_0 = 0$.

$$u_x(t_k) = v\tau\frac{x^* - x_k - \tau s(y_k)}{\lambda(t_k) - (v\tau)^2},$$

$$u_y(t_k) = v\tau\frac{y^* - y_k}{\lambda(t_k) - (v\tau)^2}$$

u_x, u_y should be recognized as boat control in moment t_k. Hence next position defined by formulae

$$x_{k+1} = x_k + \tau(s(y_k) + u_x(t_k)v),$$

$$y_{k+1} = y_k + u_y(t_k)v\tau$$

To exclude infinite looping when target cannot be reached we shall set some maximum number of iterations as $K > N$ so $k \leq K$, and if $x_k = x^*, y_k = y^*$ (target reached) then we have to define $K = k$.

Full time of reaching the target is $T^* = \tau K$ if target reached, and if not then $T^* = +\infty$.

Parameters effect research

Boat speed

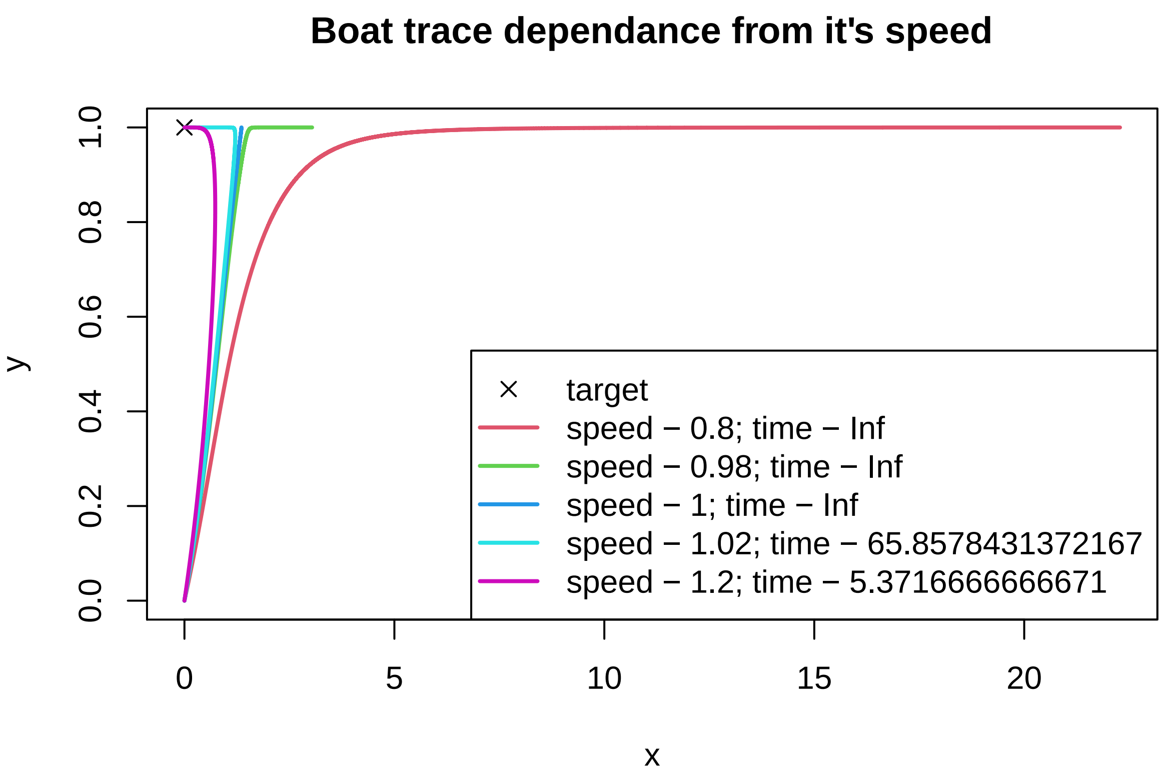
Analyzing the problem formulation and given method we can got conclusion:
Stream moves the boat only by x axis and hence must condition to reach the target is $v > s(y^*)$, $s(y)$ is continuous on interval $y \in [0, y^*]$ and $\forall y \in [0, y^*), |s(y)| < \infty$. This conclusion says that if stream speed is finite on the way then boat won't be blown into infinitely distant point from which it can't return to target. It can be moved any far away but finitely. Also conditions $s(y)$ continuous and $v > s(y^*)$ are must to boat can reach such radius of y^* where boat speed will exceed stream speed in each point on way to the target, so boat will be approaching to target on it's way. BUT it's must to say that **specified CONDITIONS are must but NOT SUFFICIENT**. It's so due to conditions doesn't provide that boat will converge those radius of y^* where $v > s(y)$. If boat wouldn't aiming to target but would move only by vertical first to reach such radius, then conditions would be sufficient. But for our case it can be the situation when $\frac{du_y}{dt} = 0$ while $t \to +\infty$, and hence $y = \int_0^\infty u_y(t)dt < y^*$, where y is boat coordinate. That's why named conditions are not sufficient.

According to formulated sentences we can modify that method next way: boat must not decrease it's vertical speed less than some defined non-zero number independently on how it close to y^* just until it reach y^*.

Now let's build some models for this method to variate speed parameter to see how boat speed affects on ability to reach target. We shall set parameters such way that it will be known stream speed in line $y = y^*$ to be able check our hypothesis. So we will fix parameters except v as next:
$s_0 = 1$,
$l = 1$,
$\phi = \pi/2$,
$f(y) = 2 - y$.
So according to formulated conditions we have: target in point $x^* = 0, y^* = 1$ and hence speed of stream decreasing from 2 to 1 while approaching. So $s(y^*) = 1$ and it is expected that boat will reach target with speed greater than 1.

We could take $f(y) = sin(\frac{y\pi}{2})$ but in such case it would be situation when stream speed wouldn't move boat from target because it's speed would be less than boat's on whole way. And boat would go to target directly.

To see influence of boat speed we will build models for set of speeds: $V = \{0.8; 0.98; 1; 1.02; 1.2\}$ and see which ones will let boat to reach target.
Traces shown on plot:

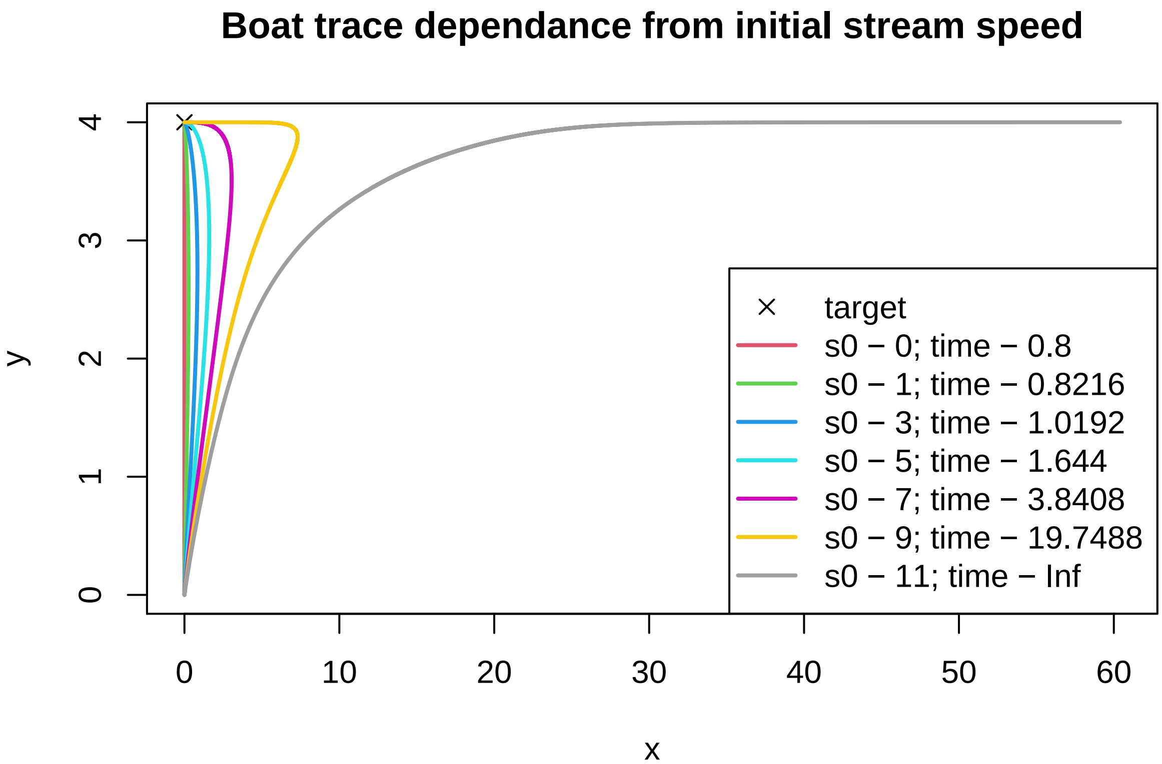


As we can see, there are 3 cases when boat can't reach target and all they are when $v \leq s(y^*) = 1$ as it was expected. In such case it is proofed to add constraint to stop algorithm when $y = y^*, x \neq x^*$ and $v \leq s(y^*)$.

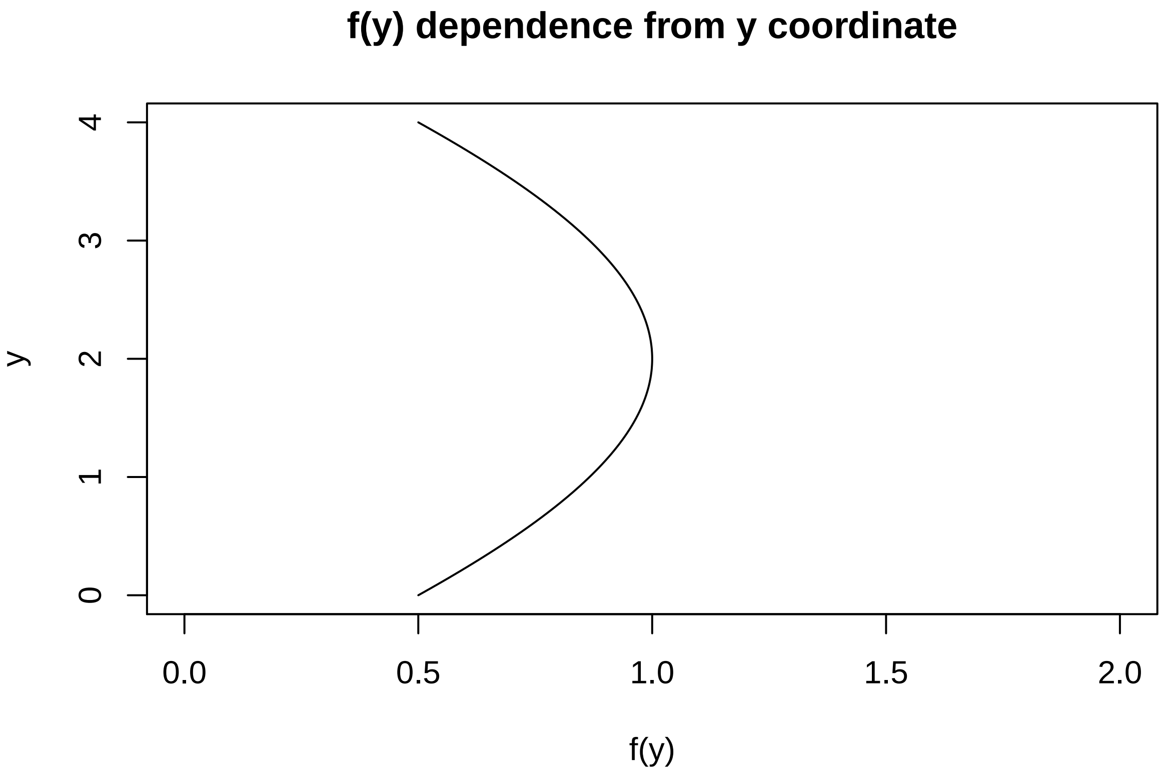
Stream initial speed

It might be not hard to make conclusion that initial speed of stream affects on time spent to reach target and probably nonlinear. Also it will affect on boat trace.

We will try to variate s_0 with next fixed values of other parameters:
$v = 5$,
$l = 4$,
$\phi = \pi/2$,
$f(y) = sin(\frac{y\pi}{6} + \frac{\pi}{6})$.
Chosen expression for $f(y)$ provides values of it between 0.5 and 1 with values 0.5 when $y \in \{0; y^*\}$. According to previous research we can variate initial stream speed as next: $s_0 \in \{0; 1; 3; 5; 7; 9; 11\}$.
Traces shown on plot:



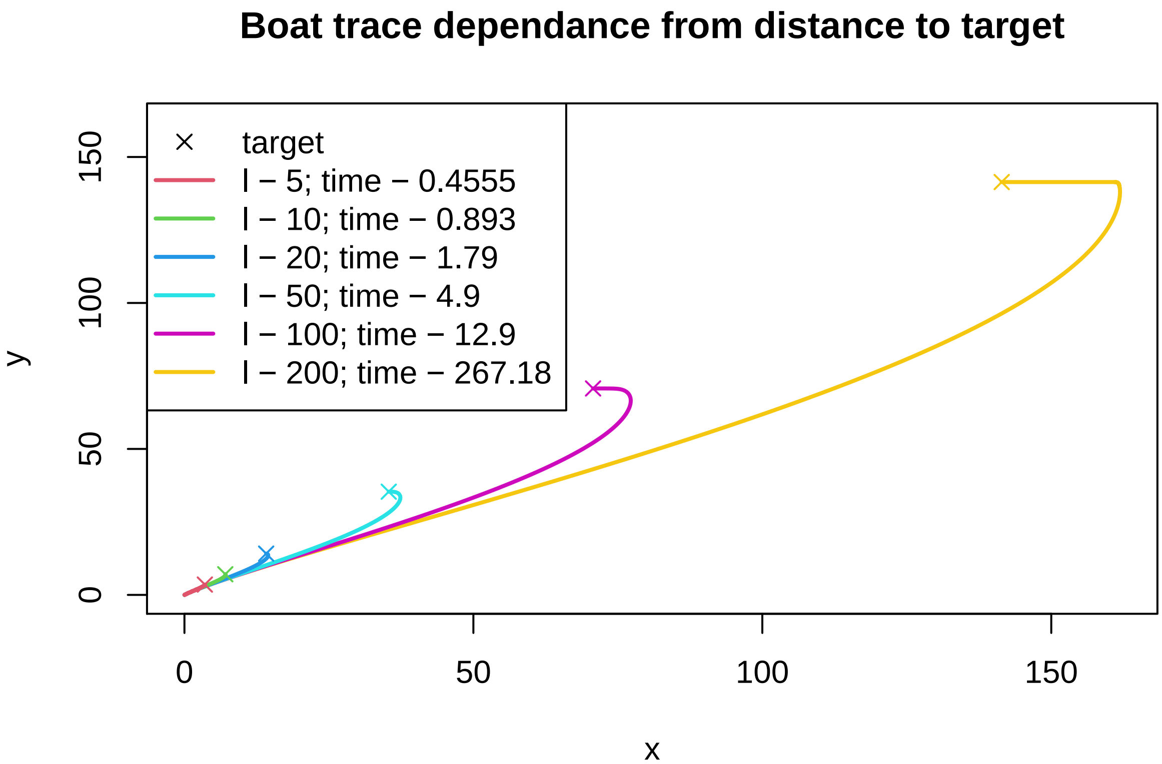
This plot may seem not very interesting, but there is one interesting moment which could be noticed earlier. But first about dependence. As we can see the time of target reach depends nonlinear on stream speed. When initial stream speed approaches to such that whole stream speed near to speed of boat the time increases much faster (the difference between time for $s_0 = 7$ and $s_0 = 9$ is much more than between $s_0 = 3$ and $s_0 = 5$). This parameter variation and previous one says that boat speed must be much greater than stream speed to reach target fast. And also in such cases stream speed affect much less on time if aiming method used. But more interesting are curves. We can see, that it's forms (especially when $s_0 = 9$) has bend near line $y = y^*$. It's interesting because we may make conclusion that boat slides vertically and moves faster to target by y coordinate at first time, but then it slows vertical move loosing speed for fighting stream speed. But we should be carefull making such conclusions. Let's see on plot of $f(y)$ speed dependence:



As we can see, it would be better to overcome middle part of way (about between 1 and 3) straight vertically to pass rapid stream as fastest possible and hence to has least loss of distance on fighting stream.

Distance to target

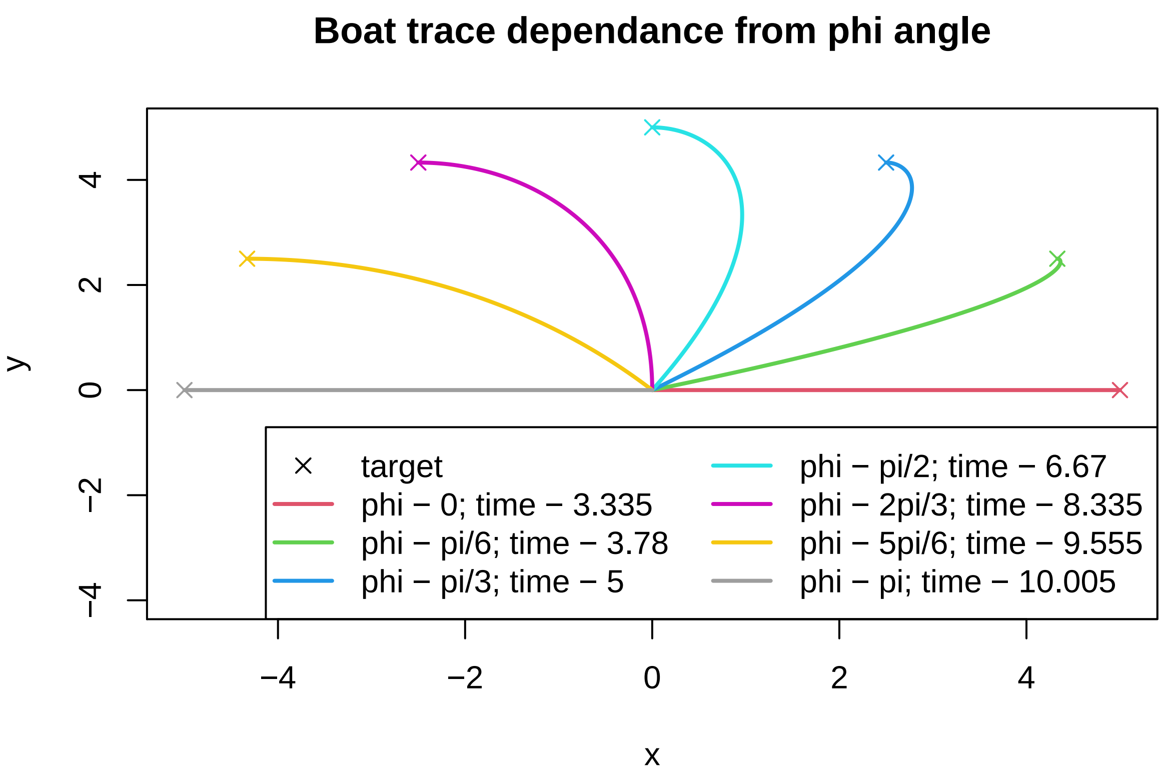
This parameter may influence on time and nonlinear but about it. To try find something interesting we will take next fixed parameters: $v = 10$ $s_0 = 2$ $\phi = \pi/4$ $f(y) = log(y+1)$ And variate $l \in L = \{5; 10; 20; 50; 100; 200\}$. As we took $\phi = \pi/4$ we expect that stream will help the boat to reach target faster on low distances.
Traces shown on plot:



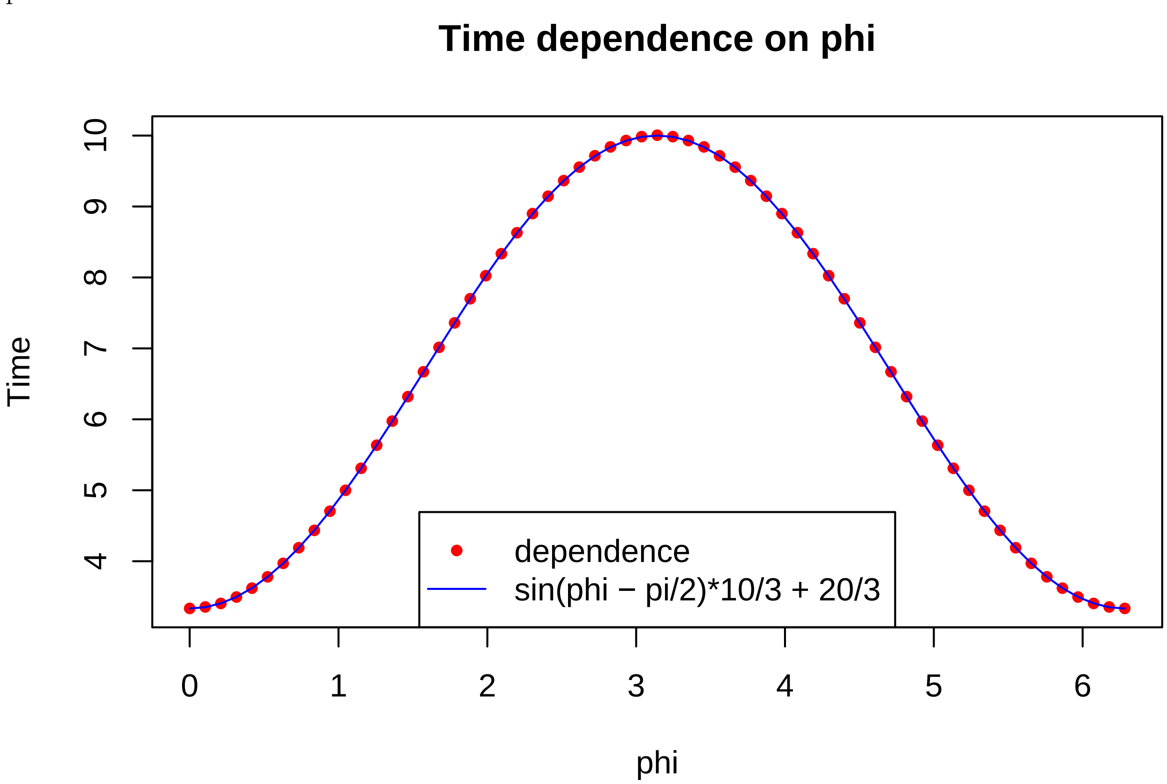
Without stream the time would be $T = \{0.5; 1; 2; 5; 10; 20\}$, but with stream we have times less than without stream for $l \leq 50$ and for $l = 10, l = 20$ there is the best profit of stream, which notably from traces (green and dark blue). Boat doesn't do a hook due to stream. In first case (red line) there is almost no profit of stream because it's speed very small on whole way. When $l = 50$ boat already made a hook but still won more time on moving by stream than lost on hook. And for more distant points $l = 100, l = 200$, boat already lost more time on hook than it got by moving along stream. Also for $l = 200$ stream speed approaches very close to boat speed: $s(y^*) = 2 * log(\frac{200}{\sqrt{2}} + 1) \approx 9.91758$, So boat lost a lot of time in this case due to reasons clarified before.

Angle between stream vector and vector from origin to target

Parameter ϕ may affect directly on time and trace but as seen in previous great influence has function $f(y)$ because it defines how much stream affects on boat. We will take each $\pi/6$ angle as ϕ in interval $[0, \pi]$. Other parameters will be next: $v = 1$ $s_0 = 0.5$ $l = 5$ $f(y) = 1$
Traces shown on plot:



It was expected that time depends on angle, it's because angle defines how much boat will move against stream. It is interesting which dependence between angle and time. To view it we will make more models for angles on $[0, 2\pi]$. It's OK that target will be lower than x axis in some cases, method allows it. Other parameters will left the same.



Dependence was such interesting and simple that I tried and built some sinus line which similar to got dependence. According to this plot we can conclude that angle almost doesn't affect on time when angle near to 0 or πk, but almost linear affect when angle around of $\pi k/4$ for $k \in \mathbb{Z}$. But as we viewed dependence when stream speed is constant the conclusion is not general.

Stream speed function

This parameter was varied through researching other parameters and we found out that it can even make target unreachable. But in fact we can prove next sentence: if $\exists \hat{y} \in [0, y^*] : |s(\hat{y})| < v$ and $\forall y \in [0, y^*] : |s(y)| < \infty$, where $s : \mathbb{R} \to \mathbb{R}$ is continuous, then boat can be reached. But it might be impossible with aiming method. Should be used other algorithm, which could be also optimal.

While researching this problem I got an idea about algorithm which could be used to provide reach and optimal control if mentioned conditions are true. I will try to prove this in next part.

Reach condition proof

Conditions says that: 1) $\exists \hat{y} \in [0, y^*] : |s(\hat{y})| < v$
2) $s : \mathbb{R} \to \mathbb{R}$ is continuous
3) $\forall y \in [0, y^*] : |s(y)| < \infty$

Every time I says "boat moves along y axis" I means it's control along y axis, so boat uses all speed to change only y coordinate and x coordinate is changed only by stream.

According to conditions 2) and 3) the function $s(y)$ is Riemann intergable on interval $[0, y^*]$ according to the theorem (about continuous function integrability). Hence according to Riemann integral properties

$$\exists I_1 \equiv \int_0^{\hat{y}} t(y)s(y)dy$$

$$\exists I_2 \equiv \int_{\hat{y}}^{y^*} t(y)s(y)dy$$

and hence

$$I = I_1 + I_2 = \int_0^{y^*} t(y)s(y)dy$$

This integral equals to whole boat shift through it's way, but it's depends on y not on time t. If boat moves directly vertically from origin to level $\hat{y}$ and after from level $\hat{y}$ to level y^* then whole shift by x axis will be I. Wherein time intervals for this move will be

$$\Delta t = \frac{\Delta y}{v} \Leftarrow (v = \frac{\Delta y}{\Delta t} = \frac{y_2 - y_1}{t_2 - t_1} = const)$$

so whole time only for pure vertical move will be:

$$T_{vrt} = \frac{\hat{y} - 0}{v} + \frac{y^* - \hat{y}}{v} = \frac{y^*}{v}$$

. And while moving vertically we have $y(t) = vt$, so $dy = vdt$. So now we can say that:

$$I = v \int_0^{\frac{y^*}{v}} ts(vt)dt$$

But there is some issue: boat won't move it's whole way by y coordinate. The idea is to move boat to such y coordinate where absolute stream speed will be less than boat's speed so that boat could move by x axis in any way. Got formula for I will just help us to know how much and whereto move boat by x axis to piece out shift of stream got while vertical move. And got I value is directly those value of x shift that boat have to piece out. We have satisfy condition: $I + \Delta x = x^*$ or simpler $\Delta x = x^* - I$, where Δx is distance which boat have to overcome along x axis to shoot into target on end of move along y axis. let's define direction as variable $d \in \{-1; 1\}$, so we have

$$\Delta x = (s(\hat{y}) + d \cdot v)T_{hor} = x^* - I$$

According to condition 1): if $d = -1$ then $\Delta x \leq 0$, else if $d = 1$ then $\Delta x \geq 0$ and $\Delta x = 0$ only if $T_{hor} = 0$. According to all mentioned about direction and Δx we have next rule:

$$d = B_-^+(\Delta x) = B_-^+(x^* - I)$$

where B_-^+ is bipolar step function which frequently used in Machine Learning. B_-^+ is my own designation which I hope is intuitive. $B_-^+(x)$ is -1 if $x < 0$ and is $+1$ if $x \geq 0$. The reason why we get bipolar step function instead of signum function is that bipolar step function doesn't return zero. Zero could cause division by zero in future. And now we can get whole time of horizontal boat move:

$$T_{hor} = \frac{x^* - I}{s(\hat{y}) + vd}$$

It could be situation, when, for example, $s(y) \equiv 0, x^* = 0$ so in such case stream shift would be zero, stream would be zero and hence we would have $\frac{0}{0}$ if we taken signum function instead of bipolar step function. And as $v \neq s(\hat{y})$ so in such case we would have zero division by some non-zero value. And it would be nice, because boat wouldn't move horizontally as it wouldn't need to.

Also whole boat move time as:

$$T = T_{vrt} + T_{hor} = \frac{y^*}{v} + \frac{x^* - I}{s(\hat{y}) + vB_-^+(x^* - I)}$$

Boat trace will be next:
at first boat moves directly vertical to reach line $y = \hat{y}$ for a time equals to $\frac{\hat{y}}{v}$ and it turns up in point $(v \int_0^{\frac{\hat{y}}{v}} t \cdot s(vt)dt; \hat{y})$. After that boat moves along x axis for a time T_{hor} and appears in point $(x^* - v \int_{\frac{\hat{y}}{v}}^{\frac{y^*}{v}} t \cdot s(vt)dt; \hat{y})$. After that boat moves again along y axis for a time $\frac{y^* - \hat{y}}{v}$ and shots directly to point (x^*, y^*), **which was to be proven**.

Algorithm proposition

As we found out next expression for the whole time:

$$T = \frac{y^*}{v} + \frac{x^* - I}{s(\hat{y}) + vd}$$

we can see that it depends on single variable $\hat{y}$ because all other values are constant in it, while due to $s(y)$ continuity there are many points which can be got as $\hat{y}$. So the idea is in next: at first we should find value I by using numeric integration. Then we must find out point $\hat{y}$ as optimal point. And it can be maximum or minimum. To recognize which optimum to find we should know value for $d = B_-^+(x^* - I)$ and if $d = -1$ then we have to find minimum to provide positive value of T_{hor} and else if $d = +1$ then we have to find maximum. Also such optimums will provide the greatest absolute value into divisor which will minimize whole fraction. So we have to find value $\hat{y}$ as mentioned optimum and after find the time for horizontal move of boat T_{hor}. To build boat trace we have to make model with vertical control to level $y = \hat{y}$, then with horizontal control in direction defined as d and again with vertical control until reaching level $y = y^*$. It might be problem due to numeric methods used: boat might end not in directly target point. In such case if stream not faster than boat in target point than it could be pieced out when boat reaches level $y = y^*$.

Proposed algorithm step by step

Set big number N. Split interval $[0, y^*]$ into N intervals of same length. Having $y_i = i \cdot \frac{y^*}{N} = i \cdot \Delta y, i = 0 \cdots N$ define $t_i = i \cdot \frac{\Delta y}{v} = i \cdot \Delta t$.
Set $x_0 = 0, y_0 = 0$.
Loop over $j = 0 \cdots N - 1$:
$x_{j+1} = x_j + \Delta t \cdot s(t_j)$;
$y_{j+1} = y_j + \Delta y$.

After loop ends set $I = x_N$ and $\Delta x = x^* - I$.
If $\Delta x = 0$ then END with got trace through points $(x_i, y_i), i = 0 \cdots N$;
If $\Delta x < 0$ then find $\hat{y} = argmin_{y \in [0, y^*]} s(y)$ and set $d = -1$.
If $\Delta x > 0$ then find $\hat{y} = argmax_{y \in [0, y^*]} s(y)$ and set $d = +1$;

Find such number $k, 1 \leq k \leq N$ that $y_{k-1} \leq \hat{y} \leq y_k$, in simple words, find the splitting interval which contains $\hat{y}$.

Then find value $\hat{x} = x_{k-1} + (x_k - x_{k-1}) \frac{\hat{y} - y_{k-1}}{y_k - y_{k-1}}$.

Loop over $j = N \cdots k$ (descending order):
set indexes shift (thus looping on descending order) and boat shift by x axis
$x_{j+2} = x_j + \Delta x$,
$y_{j+2} = y_j$

And after loop there is last hatch.
Set $x_k = \hat{x}, y_k = \hat{y}$ and
$x_{k+1} = \hat{x} + \Delta x, y_{k+1} = \hat{y}$.
Done.

Time of boat move should be calculated as such:

$$T = \frac{y^*}{v} + \frac{\Delta x}{s(\hat{y}) + vd}$$

That's all.

Proposed algorithm test and comparison

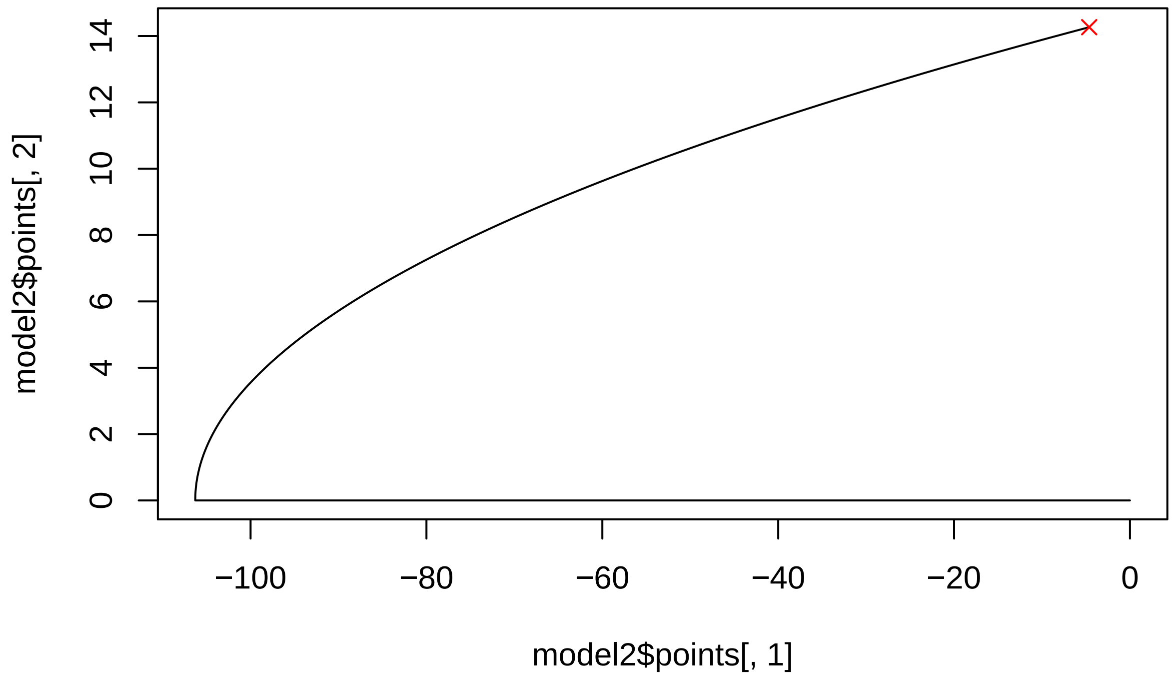
Algorithm written on R language is next:

```
my.model <- function(input, N = 1e3){
    v <- input$v
    s0 <- input$s0
    l <- input$l
    phi <- input$phi
    f <- input$f

    target <- l * c(cos(phi), sin(phi))
    s <- function(y) s0 * f(y)

    y <- 0:N * target[2] / N
    dy <- target[2] / N
    t <- y/v
    dt <- dy/v

    x <- 0
```

```r
    for (j in 1:N){
        x <- c(x, x[j] + dt*s(y[j]))
    }

    I <- x[N+1]
    Dx <- target[1] - I

    T_ver <- t[N+1]

    if (Dx == 0) return(list(points = cbind(x, y),
                             time = T_ver,
                             target = target))
    if (Dx < 0){
        d <- -1
        y_hat <- optimize(f, c(0, target[2]))$minimum
    }
    if (Dx > 0){
        d <- +1
        y_hat <- optimize(f, c(0, target[2]), maximum = TRUE)$maximum
    }

    k <- which((y[1:N] <= y_hat) & (y_hat <= y[(1:N)+1]))

    x_hat <- x[k] + (x[k+1]-x[k])*(y_hat-y[k])/(y[k+1]-y[k])

    T_hor <- Dx / (s(y_hat) + v*d)

    return(list(points=rbind(cbind(x[1:k], y[1:k]),
                             c(x_hat, y_hat),
                             c(x_hat+Dx, y_hat),
                             cbind(x[(k:N)+1]+Dx, y[(k:N)+1])),
                time = T_ver + T_hor,
                target = target))
}
```

Test it on parameters for my laboratory variant (aiming method wasn't provide target catch).

```
## Input:

## $s0
## [1] 3.872983
##
## $v
## [1] 3.872983
##
## $l
## [1] 15
##
## $phi
## [1] 1.884956
##
## $f
## function (y)
## y
```




```
## Time is 31.12878
```