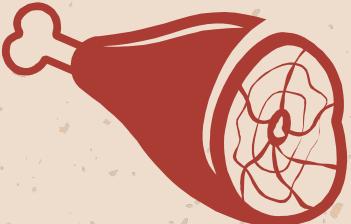


Nonlinear Approach To Searing Steak

Elliot Weiner



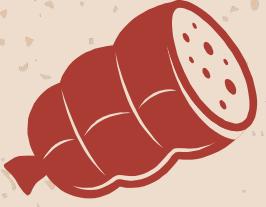
Doneness

Temperature ranges for different cooks of steak

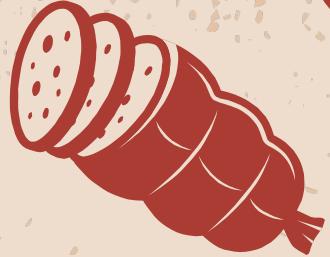
Rare	120-125°F
Medium Rare	130-135°F
Medium	140-145°F
Medium Well	150-155°F
Well Done	160-165°F
Overcooked	160°F >

For more info:

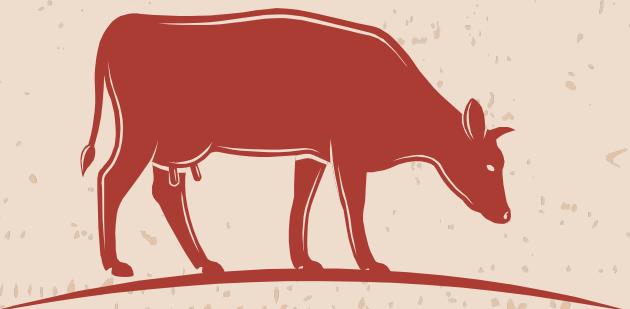




1



General Approach



Background

Cooking the perfect steak has always been an elusive goal for many cooks

It is both a science and an art, making it an interesting opportunity to apply nonlinear control



Figure 1 [2]

Previous Work



[3]

Development and
Validation of a
Computational Model for
Steak Double-Sided Pan
Cooking



[4]

A Mathematical Model for
Meat Cooking Based on
Polymer-Solvent Analogy



[5]

Extreme Steak: Wild and
Crazy Ways to Get a Killer
Sear





New Approach

Approach real-world problem with theoretical understanding

1. Derive approx. equations for
 - Internal temperature
 - Searing Index
2. Gather data across different temperatures and times
3. Nonlinear param. estimation
4. Model as nonlinear system
5. Derive control to approach goal temp/sear at time t_{final}



Table of contents

01

General
Approach

02

Internal
Temperature

03

Searing
Index

04

System
Representation

05

Finding
Constants

06

Control
Derivation

Table of contents

07

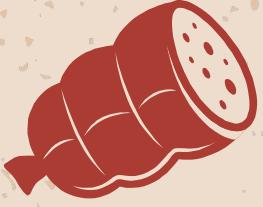
Results

08

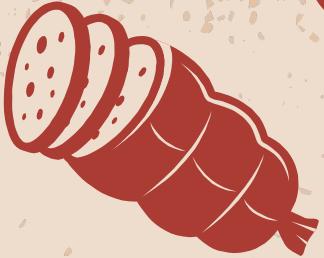
Conclusions &
Future Work

09

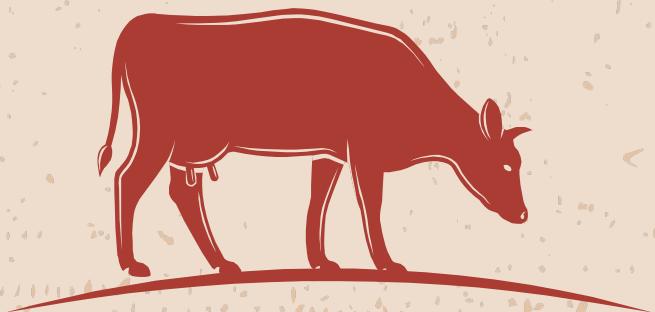
Sources



2



Internal Temperature





Modeling Temp

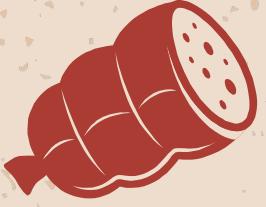
Based on broad research and intuition

- Temperature change starts slowly
- Speeds up over time
- Finally, it slows down again

It could be approximately estimated as a sigmoid [3].

$$T_{int}(T, t) \approx T_{init} + \frac{(T_{pan} - T_{init})}{1 + e^{-k \frac{t+c}{thickness}}}$$

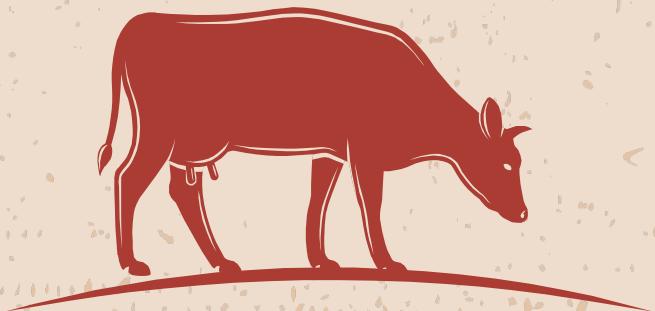




3



Searing Index





Modeling Sear

Searing starts much more initially

- Directly exposed to heat, instant browning
- Searing hits an upper limit (asymptote) where it doesn't get any darker (burnt)

Might be modeled effectively using an exponential

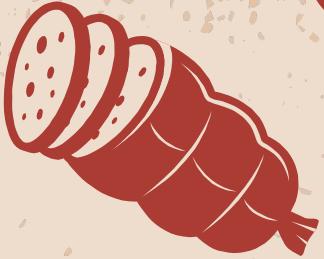
- Base EQ. on color (visual inspection)

$$C(T, t) \approx C_{max} * (1 - e^{-a(T_{pan}-b)t})$$

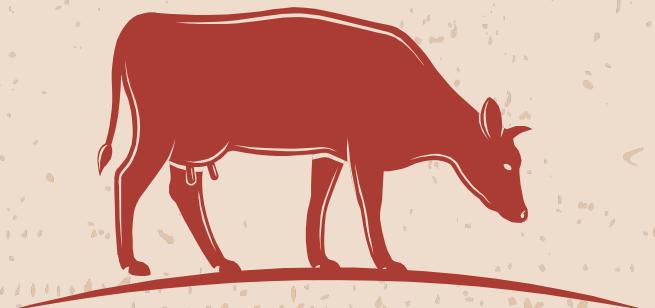




4



System Representation



Initial System

X₁

Ẋ₁

$$\widehat{X}_1 = \widehat{C} = 100a(T_{pan} - b + ut) \times e^{-a(T_{pan}-b)t}$$

$$X_1 = C = 100(1 - e^{-a(T_{pan}-b)t}) = \int_0^t \widehat{X}_1 dt$$

X₂

Ẋ₂

$$\widehat{X}_2 = \widehat{T}_{int} = u\sigma(H(t)) + cT_{diff}\sigma(H(t))(1 - \sigma(H(t)))[d + fT_{diff} + fu(t - g)]$$

Where: $H(t) = c(d + f(T_{diff}))t - g$, $T_{diff} = T_{pan} - T_{int init}$

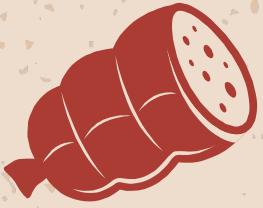
$$X_2 = T_{int} = T_{int init} + \frac{T_{pan} - T_{int init}}{\frac{-c(d+f(T_{pan}-T_{int init}))t-g}{1+e}} - \frac{T_{pan} - T_{int init}}{\frac{c(d+f(T_{pan}-T_{int init}))g}{1+e}} = T_{int init} + \int_0^t \widehat{X}_2 dt - \frac{T_{pan} - T_{int init}}{\frac{c(d+f(T_{pan}-T_{int init}))g}{1+e}}$$

Explanation

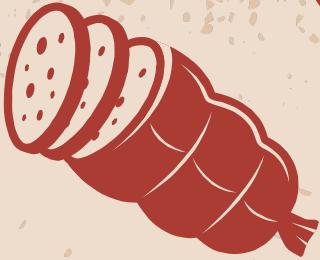
X₁ equation and derivative for sear control.

Explanation

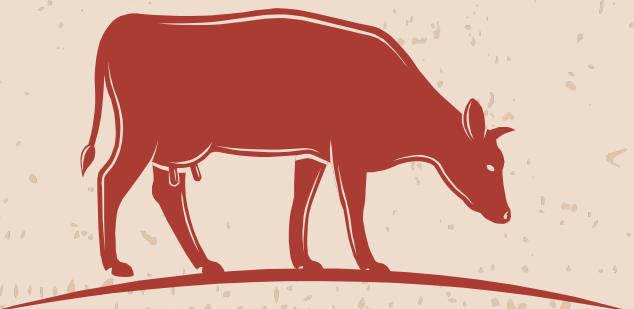
X₂ equation and derivative for internal temp control



5



Finding Constants



Experiment Setup

- Grill 1in thick steaks cut 1 in³
- Thermometer probe for internal temp (check every 10 s)
- Thermometer heat gun for pan temp (verify stable temp every 30 s).
 - Assume gaps are linear.
- Visual inspection for sear (Check every 30 s)



Figure 2

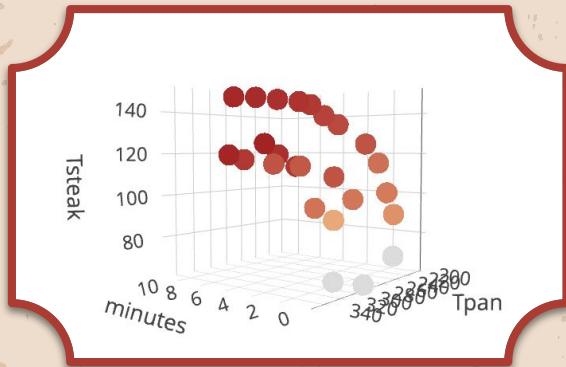


Figure 3

Internal Temp

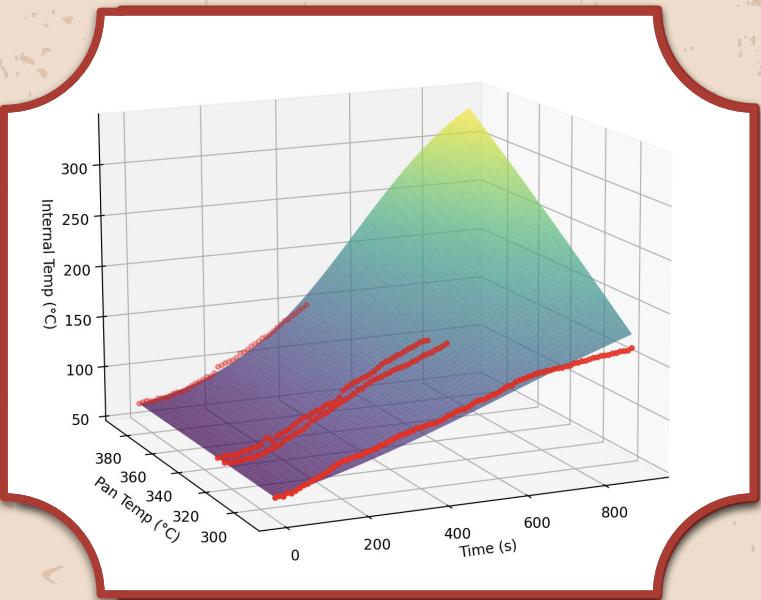


Figure 4

$$T_{int} = T_{init} + \frac{T_{pan} - T_{init}}{1+e^{-a(b+c(p-T_{init}))(t-d)}} - \frac{T_{pan} - T_{init}}{1+e^{a(b+c(p-T_{init}))d}}$$

Init time-dep. time-indep.

$$a = 1.1$$

$$b = -3.908e-03$$

$$c = 2.742e-05$$

$$d = 587.7$$

$$T_{init} = 66\text{ }^{\circ}\text{F}$$

In reality, the function required:

- a time independent offset and
- extra variables

Searing Index

$$C(T, t) \approx C_{\max} * (1 - e^{-a(T_{pan}-b)t})$$

$a = 3.315151e-05$
 $b = 170.604$
 $C_{\max} = 100$

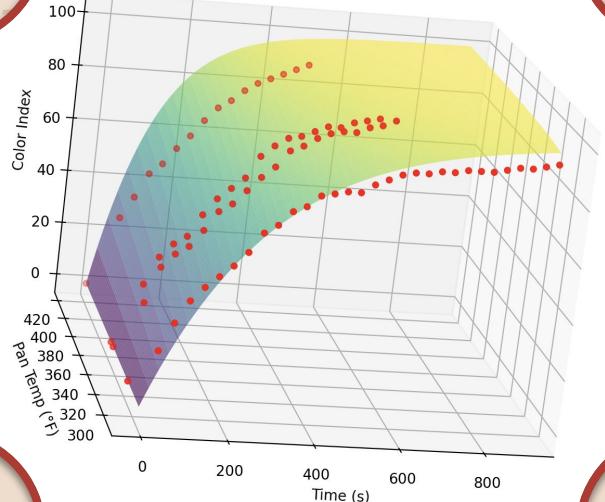


Figure 5

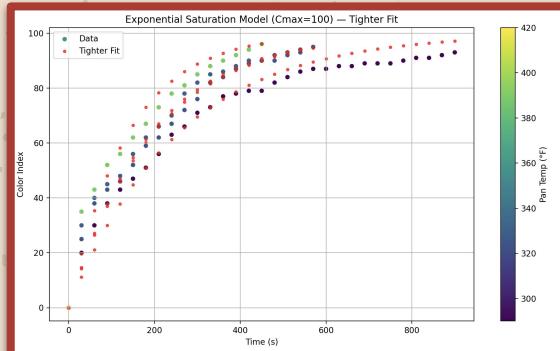
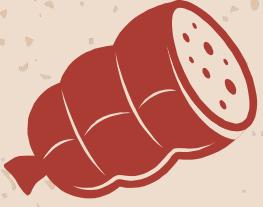
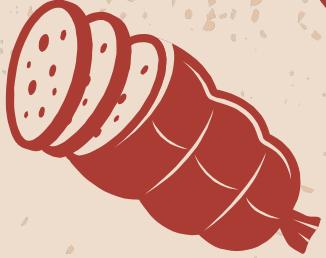


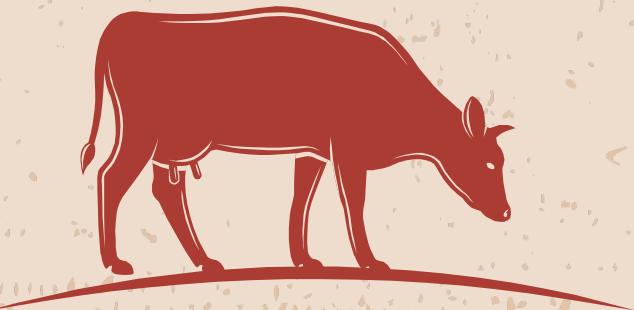
Figure 6



6



Control Derivation



Initial System

$$X_3 = T_{pan}$$

$$X_3 = T_{pan} = \int_0^t \hat{X}_3 dt + T_{pan init}$$

$$\dot{X}_3 = \check{T}_{pan}$$

$$\hat{X}_3 = \check{T}_{pan} = u$$

Explanation

Because we can control the heating rate of the pan, we can T_{pan} as a state variable and introduce controller u to the system.

Discretized System

$$x_{1,k+1} = x_{1,k} + \dot{C}(x_{3,k}, t_k) \Delta t$$

$$x_{2,k+1} = x_{2,k} + \dot{T}_{\text{int}}(x_{3,k}, t_k) \Delta t$$

$$x_{3,k+1} = x_{3,k} + u_k \Delta t$$



How do chefs do it?

Generally, a combination of PID control and feed-forward scheduling is done. Chefs use three main combinations of these:

Forward Sear

Reverse Sear

Static Temperature

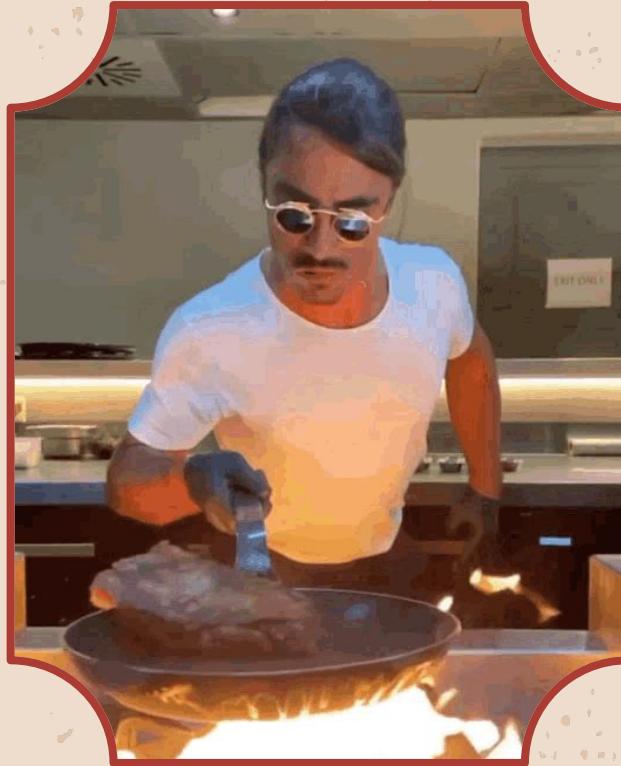


Figure 7
[\[https://tenor.com/view/saltbae-salt-bae-steak-cooking-gif-18552645\]](https://tenor.com/view/saltbae-salt-bae-steak-cooking-gif-18552645)

Nonlinear Optimal Control



Find Optimal Temp

Pre-calculate optimal temperature based on system representation. Find matched finish time for sear and internal temp.



PID as Control

Use a PID to control temperature, as convection, heat transfer, etc remove heat from the system.

Simulation

Capture ‘measurements’ by discretizing system at much smaller time steps (approx 10 ms).

Capture system data and adjust u at larger time steps for approximation (1 second).



Simulation

Capture ‘measurements’ by discretizing system at much smaller time steps (approx 10 ms).

Capture system data and adjust u at larger time steps for approximation (1 second).



```
def temp_PID_controller():
    k=(0.1, 0.005, 0.001)
    pid = PID(*k)

    return pid

def cost(Tpan, Tint, C, Tinit, x_meas):
    return abs(t_core(Tpan, Tint, Tinit, x_meas[1]) - t_crust(Tpan, C, x_meas[0]))

def optimal_pan_temp(targets, t, x_meas, bracket=(250.0, 450.0), xatol=1e-3):

    # unpack targets
    Tgoal    = targets.Tgoal
    Cgoal    = targets.Cgoal
    Tinit    = targets.Tsteak_init

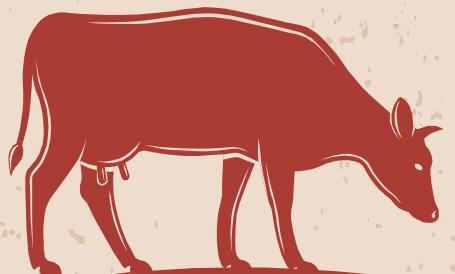
    sol = minimize_scalar(cost, bounds=bracket, args=(Tgoal, Cgoal, Tinit, x_meas), method='bounded', options={'xatol': xatol})
    print(f"optimal pan temperature: {sol.x:.1f} °F")
    print(f"residual time difference: {sol.fun:.1f} s")
    if not sol.success:
        raise RuntimeError("optimisation failed: " + sol.message)
    return sol.x, sol.fun  # (optimal Tpan, residual |Δt|)
```



7



Results





Initial Static Temperature

Matches Initial temperature

- Let the pan get up to temperature

```
optimal pan temperature: 404.1 °F
residual time difference: 0.0 s
Crust index finished: 94.92 (goal: 95.00) at time 385.0
Core temperature finished: 135.00 (goal: 135.00) at time 387.0
Ending time: 387.0 s
```

```
Crust index ended at: 95.00 (goal: 95.00)
Core temperature ended at: 135.00 (goal: 135.00)
Finished at t = 387.0 s with 388 samples.
elliotweiner@rfc1918 steak %
```



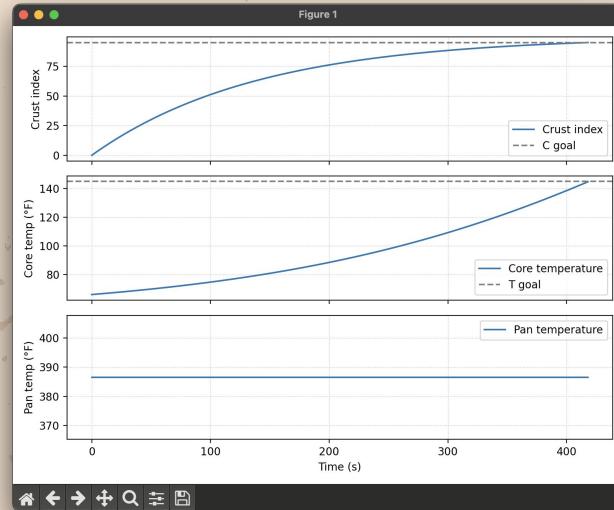
Results



Basic Simulation

Initialized system with precalculated
'perfect' temperature.

Both achieve goal in <1 sec of each other.



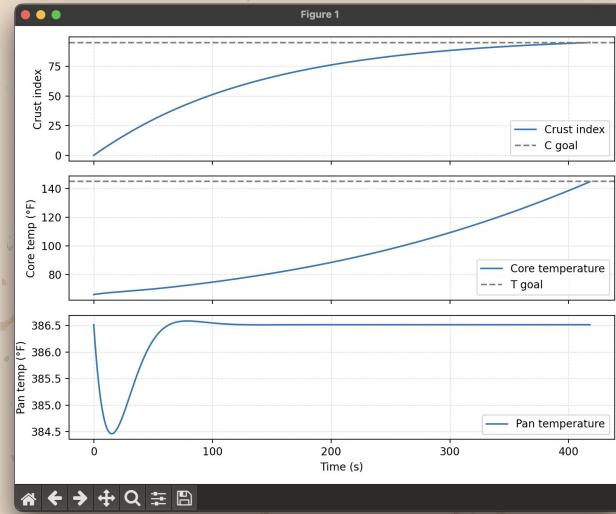
Results



Added Heat Loss

Added PID controller to counteract heat loss from pan. Robust to larger pieces of meat and convective loss.

Both achieve goal in <5 sec of each other.



Results

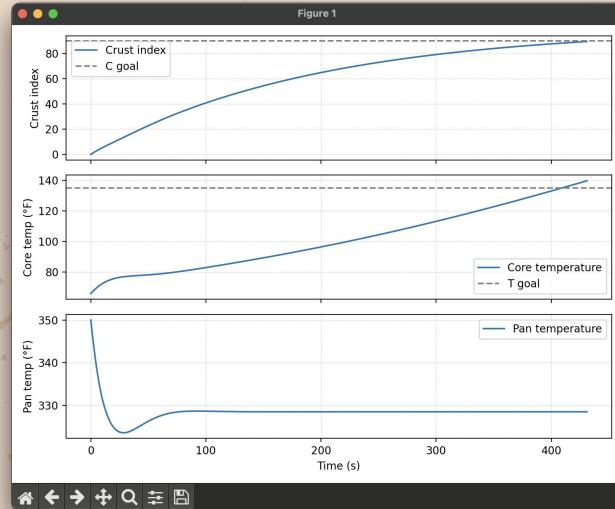


Altered Initial Temp

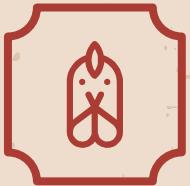
Added initial temperature offsets to show larger.

Both achieve goal in <1 min of each other at maximum temp offsets.

Potential solution: gain scheduling.



Results

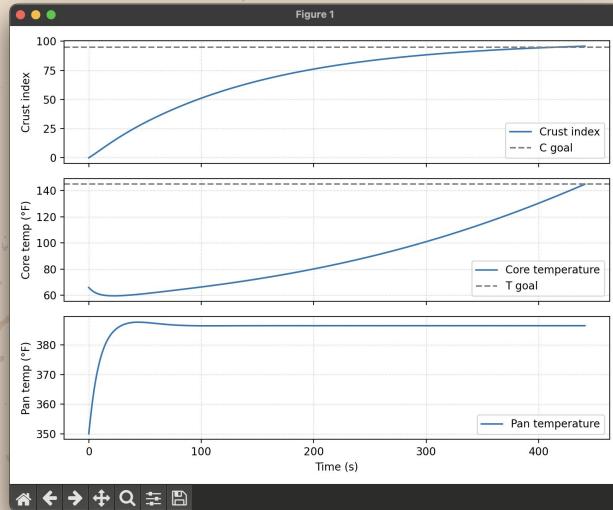


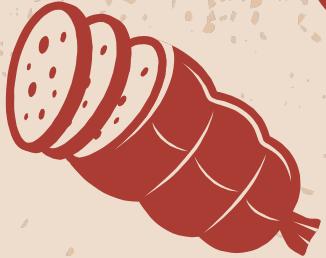
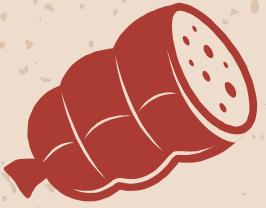
Altered Initial Temp

Added initial temperature offsets to show larger.

Both achieve goal in <1 min of each other at maximum temp offsets.

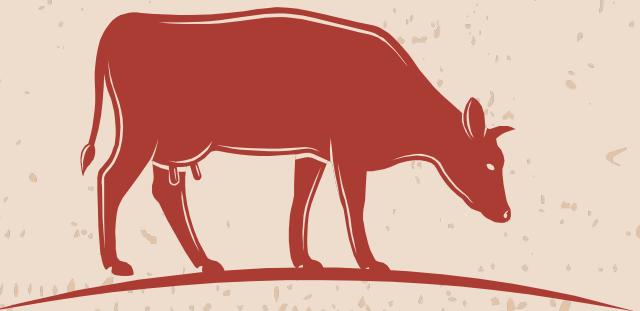
Potential solution: gain scheduling.



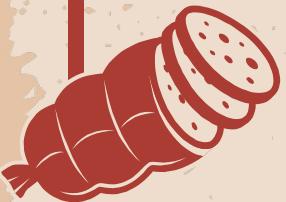


8

Conclusions & Future Work



Three ideas



Reinforcement Learning

Reinforcement learning could yield a better approximation of internal temperature, sear index, and temperature control



Better Modeling

Using Steak-unique features such as fat and muscle content, we can derive more precise control

[<https://www.goodrx.com/health-care-access/digital-health/smart-scales>]



Integrated System

With the right resources, it should be fairly trivial to build an embedded system to control steak searing



Control Graveyard

Dual PID control

- Dynamic gain scheduling
- Achieve forward/reverse sear



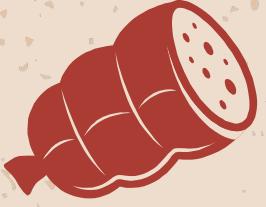
MPC

- Leverage internal model
- Lookahead (horizon)



Online Temp Planner with PID control

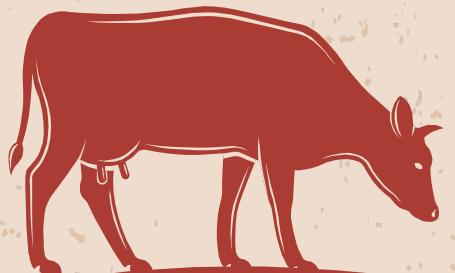




8



Sources



Resources

1. <https://ruthschris.net/blog/steak-temperature-tips/>
2. <https://searedandsmoked.com/reverse-seared-ribeye/>
3. <https://www.sciencedirect.com/science/article/pii/S0260877421000236>
4. <https://www.sciencedirect.com/science/article/pii/S0307904X14006830>
5. <https://amazingribs.com/more-technique-and-science/more-cooking-science/extreme-steak-wild-and-crazy-ways-to-kill-a-steak-killer-sear/>



