

Ivan Borisov

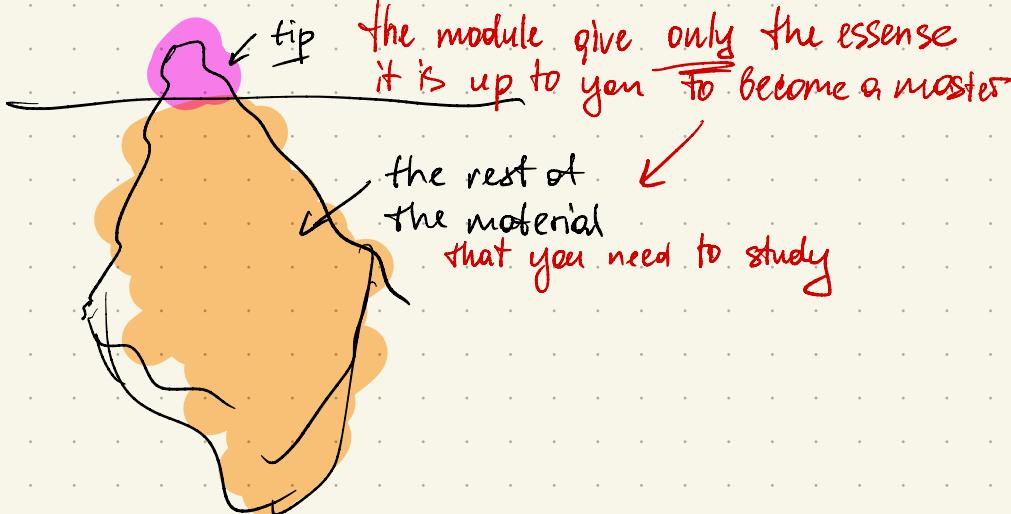
Design and Optimization
of Mechatronic Systems

Разработка и оптимизация
Мехатронных систем

Design and Optimization of Mechatronic Systems

Ivan Borisov - lectures ④

Egor Rakshin - practices ~ 12



general idea

Assume that you have a task to design a new robotic or mechatronic system.

The first question: what is the purpose?

What are the requirements?

Let us consider that our task is to create a legged robot.

1. Goal? must:

- walking
- transporting
- assembly operations
- inspection

2. Morphology

- biped
- quadruped
- with wheels?
- hopping robot

what is the purpose?

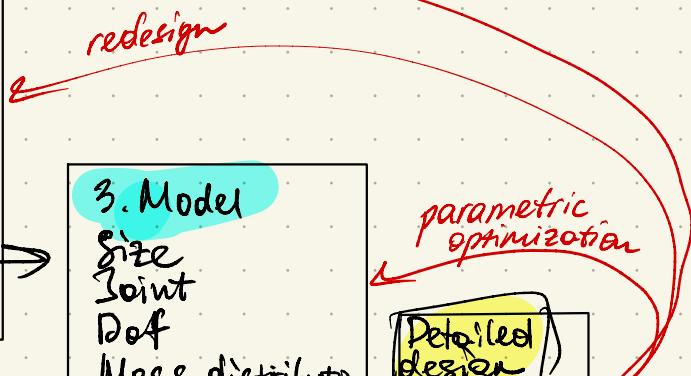
3. Model

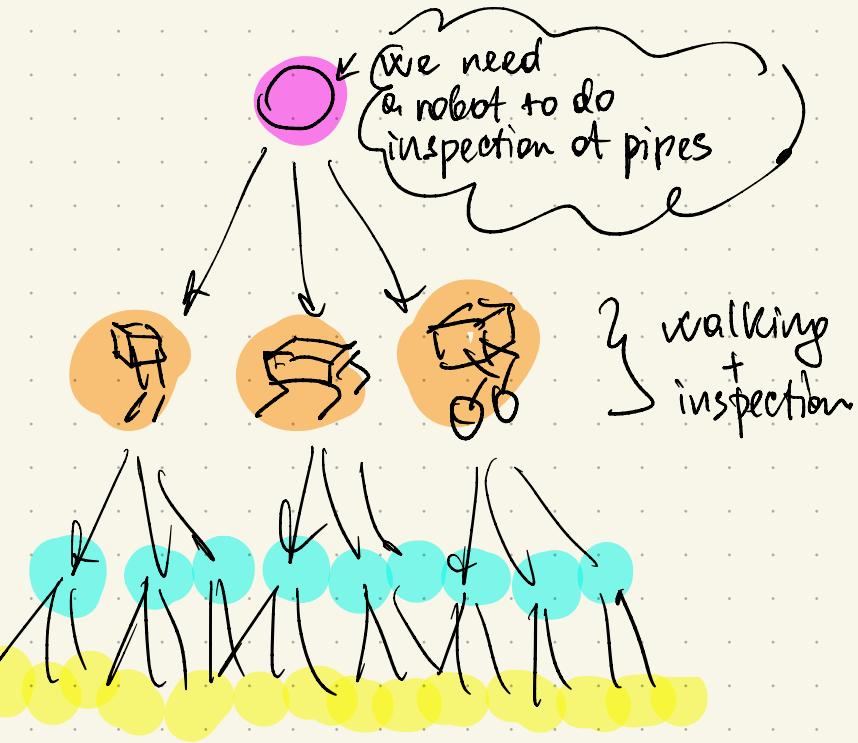
- Size
- Joint
- Dof
- Mass distribution
- ...

Detailed design

- material
- manufact
- ...

evaluate design





Design is an art as much as it is a science.
 Design is not formulated.
 There are not any strict guidelines for design.
There are no boundaries.

Computational design

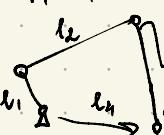
Direct design task

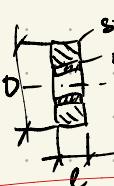
computational design treats "the creation process" as an optimization process which is "opinion-free": It just computes

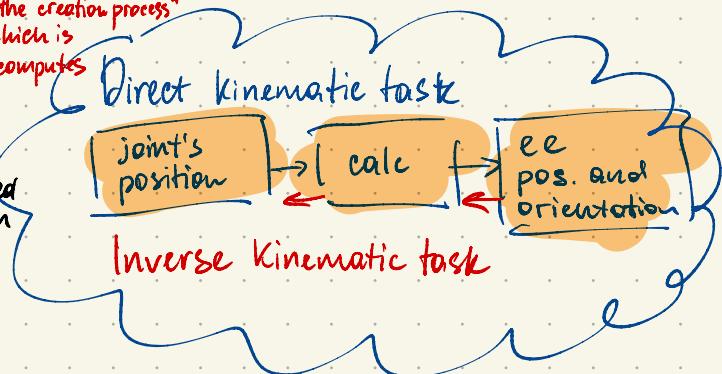
is all about human-guided evaluation of a system with initial parameters



- $k_p, k_{v_1}, k_{p_2}, k_{v_2}, \dots, k_p, k_{v_n}$: coefficients for PD

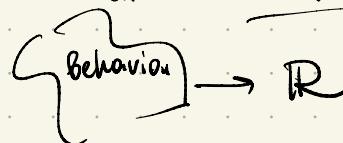
-  $\mathbf{l} = [l_1, l_2, l_3, l_n]$: geometry of the mechanics

- BLDC motor  $[D, \mathbf{l}]$: physical parameters of a BLDC.



We need to evaluate behaviour and performance numerically!!!

We need some metric!



reward function
or cost function
or fitness function

Metric that maps behaviour into real number!!!

Inverse design task

is an optimative procedure to find parameters to get the best possible design.

Разработка и оптимизация мехатронных систем

- Какое-то улучшение
- Меньшие массы
- Меньше кол-во приводов
- Адекватные термопотребление
- Отличающиеся алгоритмы (0)
- Оптимизируя СХ
- Ранергетическое звено

частные случаи что что
преследуются при разработке.
если наша задача что-то
сделать, и есть некая задача, то
это "крайний" существующее
решение.

Возможное проектирование
необходимо искать наилучшее
улучшение существующих
ищущих систем или
создать новую, с ещё большей
производительностью.

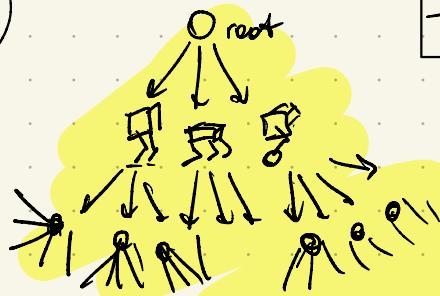
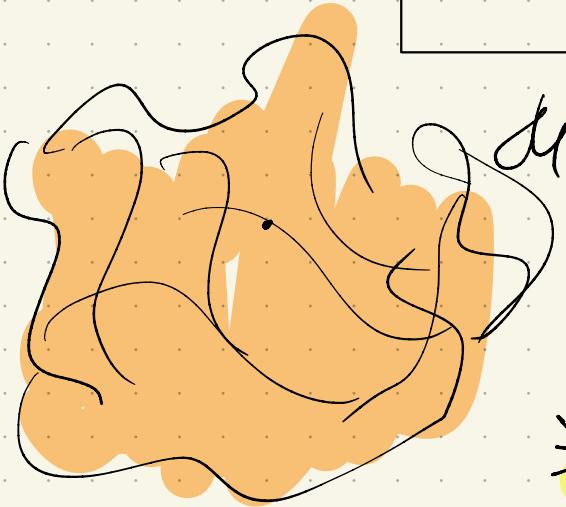
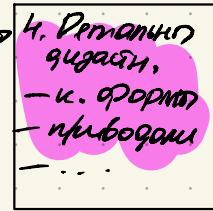
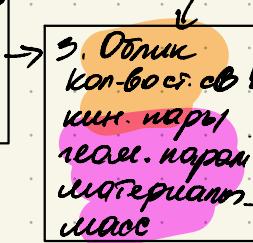
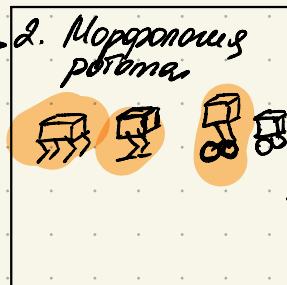
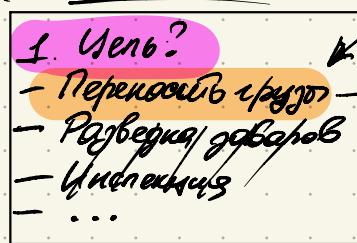
Иван Борисов: 4 пакета

Егор Ракин: 12 пакетов

Предположим есть задача строен.
иного робота.

Первый вопрос: "Для чего он нужен?"

Требования. Начиная с работы.



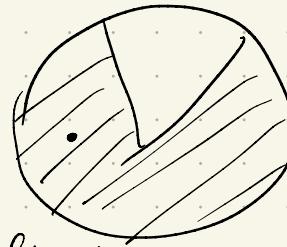
про это
мы напр.

моделирование
поведения и
производительности

Методы проектирования:

~~Биссектриса~~

ограничивает
шаги рабочего
помста,
евакуацию
и приходство



~~Реборс~~

не всегда дает наибольшие характеристики

fix

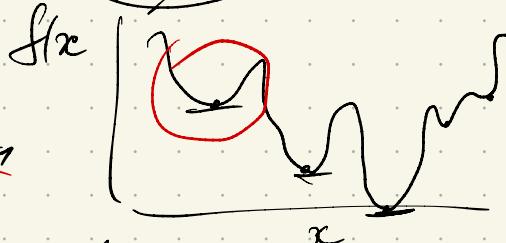
~~Генерализованное задание~~

еще сложнее

~~"Метод проб и ошибок"~~

долго и дорого

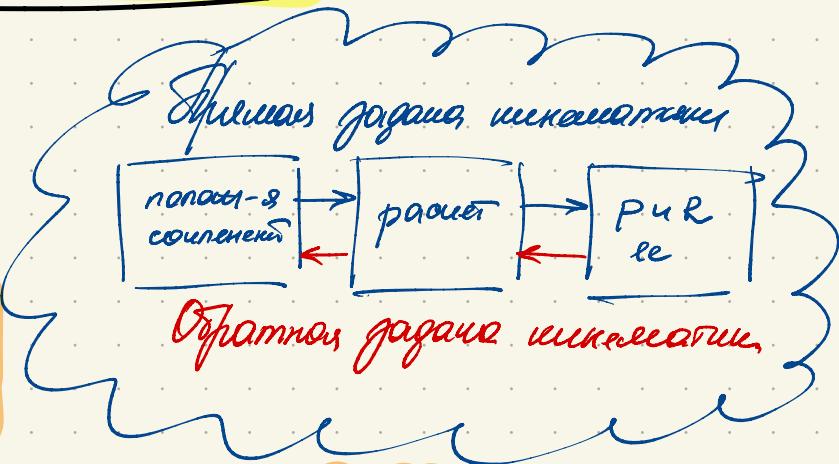
~~Низкодозимое и воспроизводимое проектирование~~



6. Волновсплесковое проектирование.

Курс об курсе об

Приемы задания проектирования.

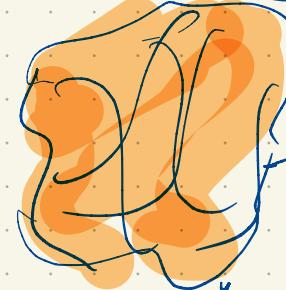


- опред. ког $\ell = [\ell_1 \ \ell_2 \ \ell_3 \dots \ell_n]$

- r, τ, γ, ω характеристики изгиболов.

- $k_p, k_v, k_{p2}, k_{v2} \dots k_{pn}, k_{vn}$

-



"Нововведение"

R

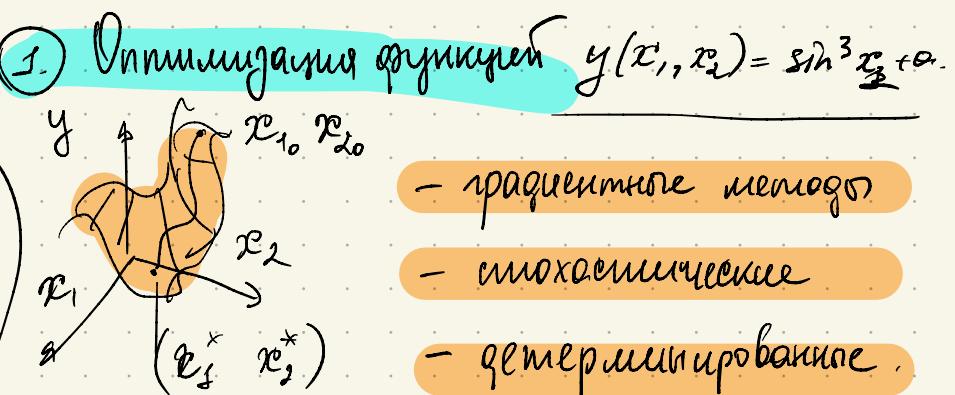
- оп-з заданы ограничения
- изменяют оп-з
- оп-з приставлены

Обратные задания проектирования

Приемы задания что нужно.

Обратные - это поиск параметров, определяющих получимую возможную систему.

- 1 семестр занят
- 2 семестр занят
- 3-й семестр происходит алгоритмы оптимизации
- 4-й семестр происходит нейросетевые алгоритмы

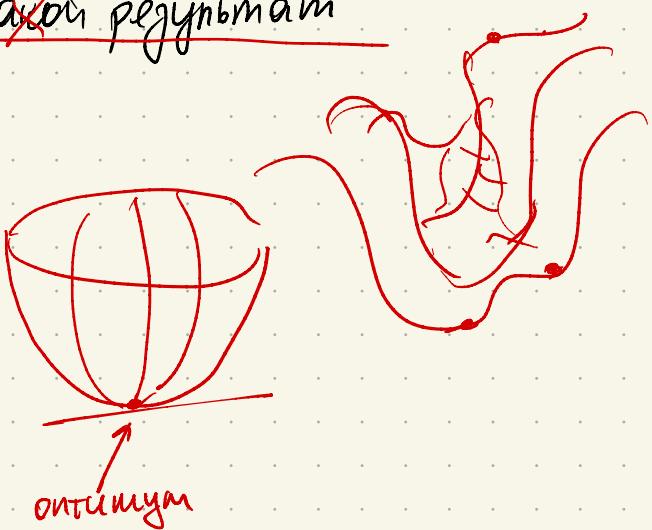
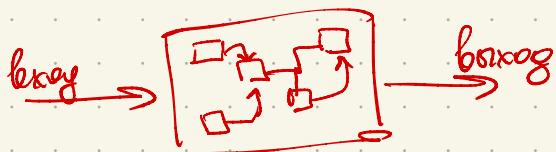


2) Проект:

- сориентировать и реализовать обратную задачу проектирования
- тема по мотивам вашей ВКР
- минимум сторонних библиотек
 - numpy
 - matplotlib
 - seirpy
 - mujoco
 - xml
- язык python

Алгоритм?

- последовательность действий и "набор регуляторов"
- все одинаковый регулятор
- набор правил "поведенческий"
- регенер, набор инструкций и правил



Что надо вспомнить постановка задачи?

- критерий
- целевая функция

- ограничения

- параметр, который хотим оптимизировать

$$Ax = b$$

$$Ax \leq b$$

$$x = [x_1, x_2] \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

СЛВ

Критерии

- время
- стоимость
- капитал
- издержки

$$t \downarrow$$

$$v \downarrow$$

$$c \downarrow$$

$$\frac{1}{\text{изд.}} \downarrow$$

$$F = w_1 t^2 + w_2 v^2$$

$$0,5 \downarrow$$

$$0,5^2 = 0,25$$

$$100^2 = 10000$$

Max.

go 8:00

ограничение на \$

А если работой?

- энергоэффективность
- масса
- спрос

Пример целевой функции

квадратичная ф-я номера дано?

$$F = \frac{1}{N} \sum_{i=1}^{N-1} (f(x) - y)^2$$

$F \in \mathbb{R}$

параметры

переменные

$$F = r_1 + r_2 + r_3$$

$$F = \frac{N}{D}$$

Алгоритм прямой силы

brute force



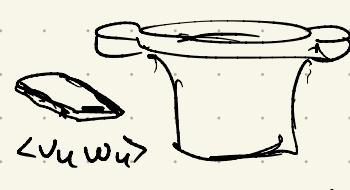
w^*



$\langle v_1 w_1 \rangle$



$\langle v_3 w_3 \rangle$



$\langle v_5 w_5 \rangle$

$V[i] = 1$ $\begin{cases} 1 & \text{если } v_j \\ 0 & \text{не } v_j \end{cases}$

нужен V при $\sum V[i] \cdot \text{value} \leq \max$.
 $\sum V[i] \cdot \text{weight} \leq w$

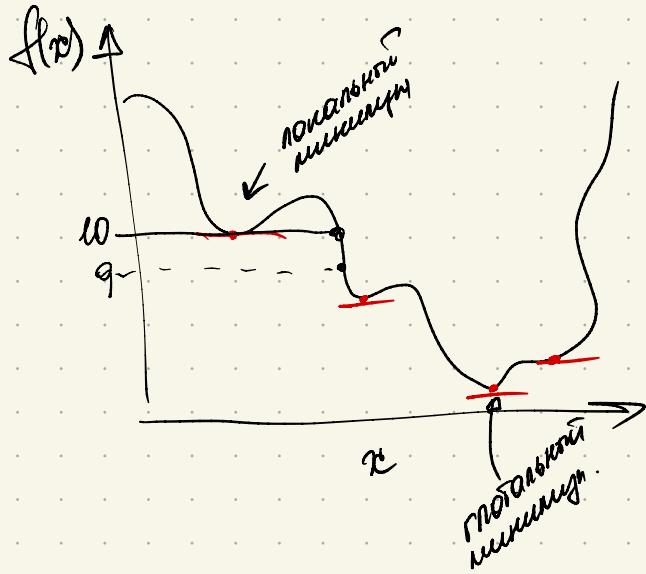
1.

2.

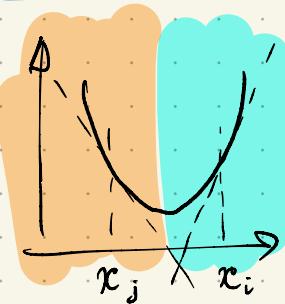
3.

4.

5.



Градиентный спуск



$$f(x_i) > 0$$

$$f'(x_i) < 0$$

$$x^* \Rightarrow f'(x^*) = 0.$$

$$x_{n+1} = x_n - d \cdot \frac{df(x)}{dx}$$

использовано однородством, но подтверждено "сваливанием" в локальной оптимум и работает только с локальными функциями

Метод имитации отыскания

стochasticный
результат
зависит от
вероятности P

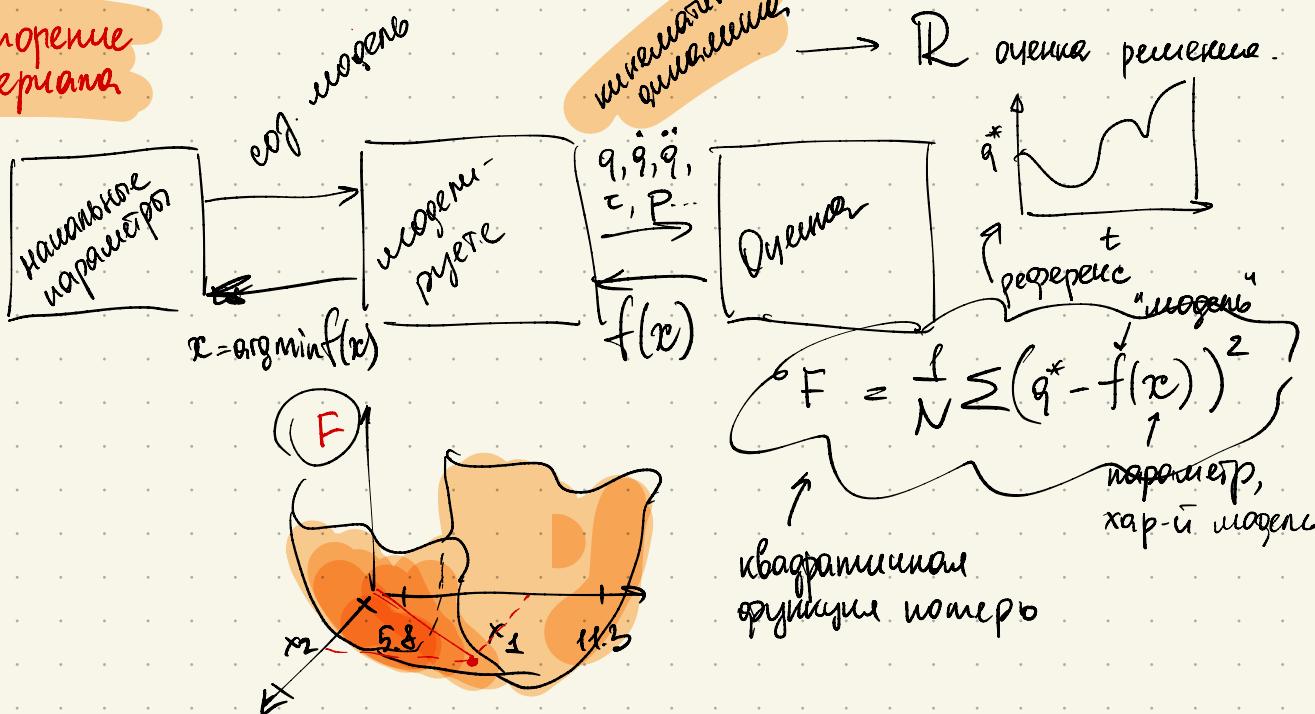
$$P(x^* \rightarrow x_{i+1} | x_i) = \begin{cases} 1 & F(x^*) - F(x_i) < 0. \\ e^{-\frac{F(x^*) - F(x_i)}{T}} & F(x^*) - F(x_i) \geq 0. \end{cases}$$

$$e^{-\frac{1}{10}} = \frac{1}{e^{0.1}} = \frac{1}{1.1} = 0.9048$$

$$e^{-\frac{5}{10}} \approx 0.6$$

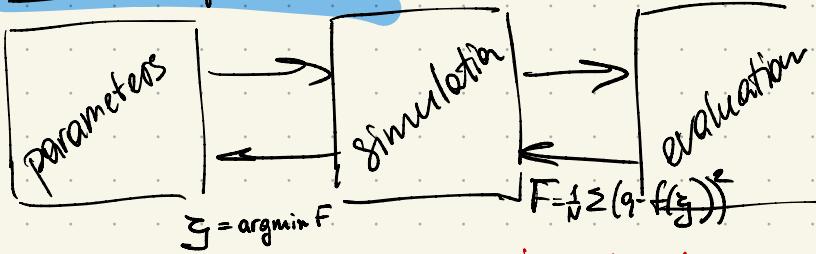
$$e^{-\frac{5}{3}} = 0,1889$$

Повторение материала



Direct Design Task

is all about evaluation

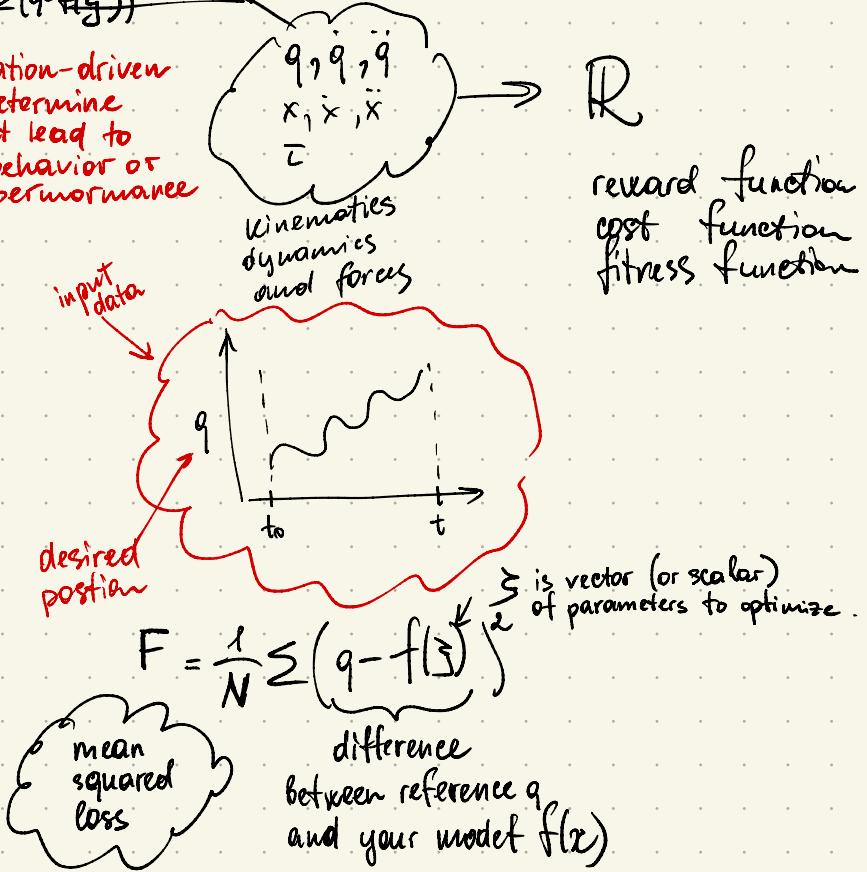


Inverse design Task

is optimization-driven task to determine values that lead to a desired behavior or performance

Algorithms for optimization

- brute force
- gradient based algorithms
- stochastic algorithms
- deterministic algorithms.



Brute force algorithm

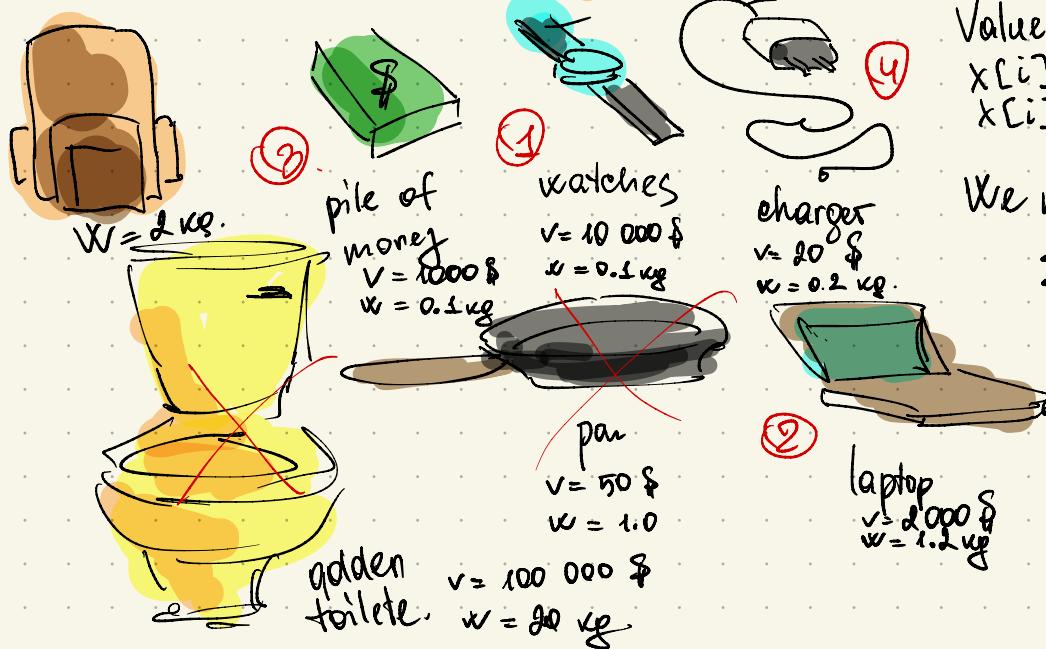
Knapsack problem

1. criteria: the biggest value

2. parameters: $x = [0, 1, 0, 1 \dots]$

3. constraints:

$x[i] = 0$ means that element is left behind
 $x[i] = 1$ means we took it.
 n is number of elements you consider



Value of each object:

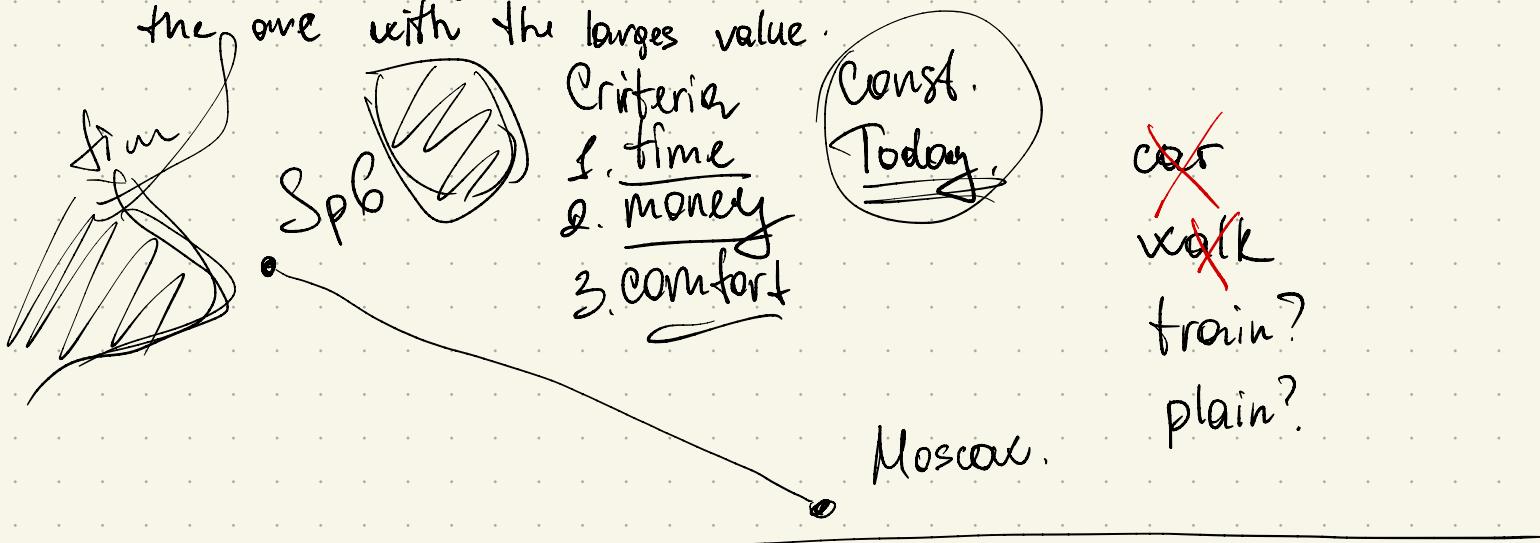
$x[i].value$
 $x[i].weight$

We need to find X s.t.

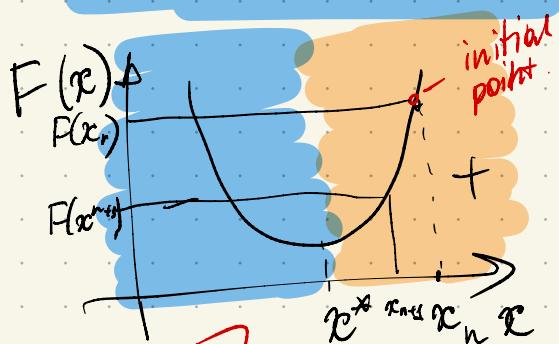
$\sum x[i].value$ is maximized.
but constraint $\sum x[i].weight \leq k$

Brute force algorithm:

1. Enumerate all possible combination of items.
2. Remove sets that do not satisfy constraints.
3. From remaining combinations choose the one with the largest value.



Gradient descent:

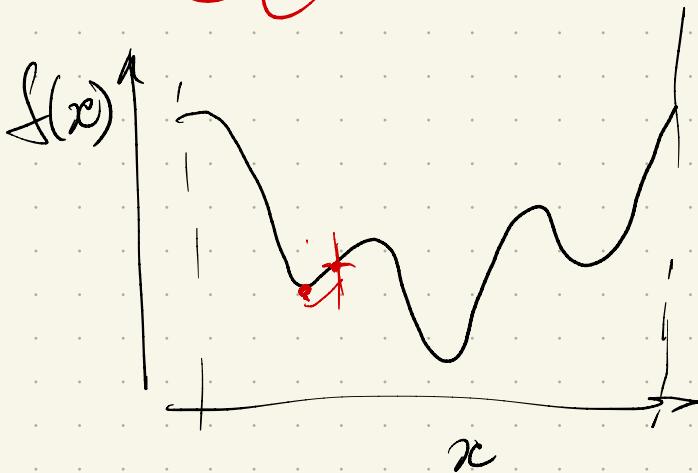


$$x_{n+1} = x_n - \alpha \frac{df(x)}{dx}$$

$$F(x^{n+1}) - F(x^n) < 0$$

- it works with only smooth functions
- issue with global solutions.

our goal is to find x^* where $F(x^*)$ has the minimal value.



Simulated annealing

$$P(x^* \rightarrow x_{i+1}(x_i)) \left\{ \begin{array}{l} 1 \quad F(x^*) - F(x_i) < 0 \\ e^{(-\frac{F(x^*) - F(x_i)}{\tau_i})} \quad \text{if } - \geq 0. \end{array} \right.$$

$$e^{-\frac{1}{10}} = \frac{1}{e^{0.1}} = \frac{1}{1.1} = 0.9048 \quad 90\%$$

$$e^{-\frac{5}{10}} = \frac{1}{e^{0.5}} = 0.6 \quad 60\%$$

$$e^{-\frac{5}{3}} = 0.1889$$