Optimization for solving certain estimation problems

Given a set of 1D constraints

$$\begin{Bmatrix} h_1(\mathbf{x}) = 0 \\ h_2(\mathbf{x}) = 0 \\ \dots \\ h_M(\mathbf{x}) = 0 \end{Bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T \in R^N$$

$$M \geq N$$

We define a scalar cost function, whose solution is the solution of the **M** individual equations.

$$C(\mathbf{x}) = \sum_{i=1}^{M} \left(h_i(\mathbf{x})\right)^2$$

$$C(\mathbf{x}) = 0 \qquad \Rightarrow \qquad \left\{h_i(\mathbf{x}) = 0\right\}_{i=1}^{M}$$

$$\left\{h_i(\mathbf{x}) = 0\right\}_{i=1}^{M} \qquad \Rightarrow \qquad C(\mathbf{x}) = 0$$

$$C(\mathbf{x}) \geq 0, \quad \forall \quad \mathbf{x}$$

We can refine that cost function, for scaling variables, or for giving different relevance to the individual constraints

$$C(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \cdot \left(h_i(\mathbf{x})\right)^2, \quad \alpha_i > 0$$

We can define other versions of valid cost functions

$$C\left(\mathbf{x}\right) = \sum_{i=1}^{M} \alpha_i \cdot \left| h_i\left(\mathbf{x}\right) \right|$$

$$C\left(\mathbf{x}\right) = \sum_{i=1}^{M} \alpha_i \cdot \left| h_i\left(\mathbf{x}\right) \right|^p$$

$$C\left(\mathbf{x}\right) = \sum_{i=1}^{M} \alpha_i \cdot g\left(\left| h_i\left(\mathbf{x}\right) \right| \right)$$

$$\left(g\left(0\right) = 0 \quad ; g\left(\mu\right) > 0; \quad \frac{dg\left(\mu\right)}{d\mu} > 0 \quad \forall \mu > 0 \right)$$

$$C(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \cdot \left| h_i(\mathbf{x}) \right|$$

$$C(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \cdot \left| h_i(\mathbf{x}) \right|^p$$

$$C(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \cdot g\left(\left| h_i(\mathbf{x}) \right| \right)$$

$$\left(g(0) = 0 \quad ; g(\mu) > 0; \quad \frac{dg(\mu)}{d\mu} > 0 \quad \forall \mu > 0 \right)$$

Any of them does satisfy

$$if \quad C(\mathbf{x}) = 0 \quad \Leftrightarrow \quad \left\{ h_i(\mathbf{x}) = 0 \right\}_{i=1}^{M}$$

Now, given a set of constraints, considering uncertainty in certain measurements , parameters and models, the problem is:

$$\left\{ h_i\left(\mathbf{x}\right) \cong 0 \right\}_{i=1}^{M}$$

$$\begin{aligned} h_1\left(\mathbf{x}\right) + \eta_1 &= 0 \\ h_2\left(\mathbf{x}\right) + \eta_2 &= 0 \\ &\cdots \\ h_M\left(\mathbf{x}\right) + \eta_M &= 0 \end{aligned}$$

$$\left\{ h_i\left(\mathbf{x}\right) \cong 0 \right\}_{i=1}^{M}$$

$$h_0\left(\mathbf{x}\right) + \eta_0 = 0$$

$$h_1\left(\mathbf{x}\right) + \eta_1 = 0$$

$$\dots$$

$$h_M\left(\mathbf{x}\right) + \eta_M = 0$$

Ideally, we would like a solution:

$$C\left(\mathbf{x}^*\right) = \sum_{i=1}^{M}\left(h_i\left(\mathbf{x}^*\right)\right)^2 = \sum_{i=1}^{M} 0^2 = 0$$

$\mathbf{x}^*$: solution

but that ideal solution may not exist.

So, we would try to achieve:

$$\left\{ h_i\left(\mathbf{x}\right) \cong 0 \right\}_{i=1}^{M}$$

$$C\left(\mathbf{x}^{*}\right) = \sum_{i=1}^{M}\left(h_i\left(\mathbf{x}^{*}\right)\right)^{2} = \zeta \rightarrow 0$$

Consequently, we try to minimize the cost function

$$\mathbf{x}^{*} = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

$$C\left(\mathbf{x}^{*}\right) \leq C\left(\mathbf{x}\right), \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}}$$

Example (familiar to us): Map-based Localization seen as an optimization problem:

Suppose our range-only localization case , in which each measured range, which corresponds to an identified map landmark, does provide a constraint.
Suppose, after processing a LiDAR scan, we detect 4 OOIs, which we are able to associate to known map landmarks. So, we estimate the solution of the trilateration problem by minimizing a cost function.

$$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$

$$\mathbf{x}^* = \underset{\mathbf{x} \in \Omega_{\mathbf{x}}}{\operatorname{argmin}} \left(C\left(\mathbf{x}\right) \right)$$

$$C\left(\mathbf{x}\right) = \sum_{j=1}^{4} \left| r_j - \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2} \right|$$

In which r_j *is a range measurement of an OOI associated to a landmark whose global position is* $\left(x_j, y_j\right)$

How to "minimize a function"?

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

Certain simple cases can be solved analytically.
Complicated cases are usually solved numerically, by "optimizers"

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

Numerical tools, for optimization

Matlab optimization toolbox.

E.g. function “fminsearch”
(which searches for “the” minimum of a function)

Function **fminsearch** can minimize functions

$$C(\mathbf{x}) \quad : \quad \mathrm{R}^N \rightarrow \mathrm{R}$$

(which is the type of function of our cost functions)

We see some example and source code in class.

Trivial example, 3D, whose solution we know (analytically).
It has a unique minimum; it is a convex function. We see how to implement it (pretending we cannot solve it analytically.)

$$\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}$$

$$C(\mathbf{x}) = (x-1)^2 + (y-2)^2 + (z-3)^2$$

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C(\mathbf{x})\right) = ?$$

(go to Matlab)

Solve the case of M range measurements.

(You solve/solved this one, I show it working in class)

$$\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$$

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

$$C\left(\mathbf{x}\right) = \sum_{j=1}^{M}\left|r_j - \sqrt{\left(x - x_j\right)^2 + \left(y - y_j\right)^2}\right|$$

Example 3: 3DoF **parameter identification**, for a SISO LTI system

We know the structure of the ODE, but we do not know the parameters a,b,c.

$$\frac{d^2 y(t)}{dt^2} = -a \cdot y(t) - b \cdot \frac{dy(t)}{dt} + c \cdot u(t)$$

Suppose have some experiments in which we measured u(t) and y(t).

For a given u(t) (which we applied , or which we know through measurements), we can simulate the output assuming that input and hypothetical values of a,b,c (and initial conditions)

So that the optimizer can use the following cost function:

…the optimizer can use the following cost function:

$$C(a,b,c)=\sum_{k=0}^{L}\left(y_m(t_k)-y_s(t_k)\right)^2$$

in which

$\left\{y_m(t_k)\right\}_{k=0}^{L}$: measurement of system's output

$\left\{y_s(t_k)\right\}_{k=0}^{L}$: simulated response based on known u(t) and initial conditions.

and on provided parameters a,b,c.

(provided by optimizer in each optimization iteration).

$$C(a,b,c)=\sum_{k=0}^{L}\left(y_m(t_k)-y_s(t_k)\right)^2$$

$\left\{y_m(t_k)\right\}_{k=0}^{L}$: measurement of system's output

$\left\{y_s(t_k)\right\}_{k=0}^{L}$: simulated response based on known u(t) and initial conditions.
and on provided parameters a,b,c.
(provided by optimizer).

In this case we do not even have an explicit expression for C(**x**), but we can evaluate it numerically, based on data and on hypothetical values of **x**.
The optimizer calls our function proposing certain **x**, we evaluate the cost function and return that result. The optimizer decides how to “navigate” in the domain of **x**, in search of the minimizing **x**.
(In our function, we can do whatever we want, even running a simulation of the ODE.)

$$C(a,b,c) = \sum_{k=0}^{L} \left(y_m(t_k) - y_s(t_k) \right)^2$$

$\{y_m(t_k)\}_{k=0}^{L}$: measurement of system's output

$\{y_s(t_k)\}_{k=0}^{L}$: simulated response based on known u(t) and initial conditions.
and on provided parameters a,b,c.
(provided by optimizer).

We see it working, in slow motion, in Matlab.

Can we tune our kinematic model?

E.g. using the approach for estimating the gyroscope bias, based on sensors' measurements during a short period of time? (e.g. 5 seconds)

Available data: we measure speed and angular rates, and LiDAR scans, during that period of time.
We are able to detect OOIs and associate some of them to known landmarks.

How would you do it?

(To be discussed in class.)

(To be discussed in class.)

Consider these two cases:
case 1) we do know vehicle initial pose. We do not know bias.

Case 2) we do not accurately know initial pose. We do not know bias.

Hint: in case 1, the optimization searches in the 1D space of “bias”

Hint: in case 2, the optimization searches in the 4D space of “bias” and “pose0”. In this case, the initial guess to the optimizer is relevant, to avoid the optimizer converging to a local minimum.

Can we tune our kinematic model?

Case 3: we have a kinematic model based on steering angle.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cdot\cos(\phi(t)) \\ v(t)\cdot\sin(\phi(t)) \\ \frac{v(t)}{L}\cdot\tan(\beta(t)) \end{bmatrix}$$

$$L = ?$$

But we do not have an accurate value for parameter L.

We do have noisy measurements of $\boxed{v(t), \beta(t)}$
and sporadic LiDAR scans, (so, we detect OOIs and associate them to known landmarks)

$$\mathbf{x}^* = \underset{\mathbf{x}\in\Omega_{\mathbf{x}}}{\operatorname{argmin}}\left(C\left(\mathbf{x}\right)\right)$$

However, the method implemented by this optimizer can converge to a local minimum (missing the global one!).

Next: We discuss one of the alternative approaches.

(we end this lecture here)

Next lecture Part B: **Particle Swarm Optimization (PSO)**