Constraints and Games

Sanjiban Choudhury
ARE WE THERE YET !?!
Linear (LQR)

Non-Linear (iLQR)

Linear (LQR)
Model-Predictive Control

- Continuously optimizes trajectory subject to nonlinear momentum dynamics
- Solve for future kinematic configurations
- Leverages optimized code and problem structure for speed
Any real world robot has to obey **hard constraints** from physics, safety, legal, ...
Activity!
Think-Pair-Share!

Think (30 sec): What are hard constraints for a self-driving car navigating an intersection?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
So …

How *do* we deal with these constraints?
Re-parameterization: The quick ’n’ easy way to solve constraints!
Example: Swing up using iLQR
How do we enforce a torque limit?

\[ \tau_{\text{min}} \leq \tau \leq \tau_{\text{max}} \]
Idea: Reformulate the variables so the constraint must be satisfied

\[ \tau_{min} \leq \tau \leq \tau_{max} \]

\[ \tau = \text{Sigmoid}(z, \tau_{min}, \tau_{max}) \]
Recipe for Re-parameterization

\[ x^* = \arg \min_x f(x) \]  
(Unconstrained objective)

Such that

\[ x \in X_{\text{feasible}} \]
Recipe for Re-parameterization

\[ x^* = \arg \min_x f(x) \quad \text{s.t.} \quad x \in X \text{feasible} \]

Step 1: Reformulate the variables so the constraint must be satisfied

\[ x = g(z) \quad \text{where} \quad z \in [-\infty, \infty] \]

Step 2: Solve the unconstrained optimization problem in \( z \)!

Step 3: Plug in \( z^* \) to get constrained optimal solution \( x^* = g(z^*) \)
Fun Fact: Dynamics is form of re-parameterization

\[ x_{t+1} - f(x_t, u_t) = 0 \]

Think about how you would deal with dynamics in a non-reparametrization fashion ...
... when does re-parameterization fail?
Failure 1: Stuck on the far side of the sigmoid

Let’s say $z$ is very high

What is $\frac{\partial x}{\partial z}$?
Failure 2: Constraints too complex to re-parameterize

\[
\begin{align*}
\min_{x} f(x) \\
\text{Such that} \\
g(x) &= 0 \\
h(x) &\leq 0
\end{align*}
\]
Hang on .... Why not put a really really really high cost for violating constraints?
Penalty method

\[
\min_{x} f(x) + \frac{\alpha}{2} g(x)^2
\]

Seems easy to implement ... what could possibly go wrong?
Activity: Apply Penalty Method!

$$2(x_1 - 4)^2 + (x_2 - 1)^2$$

s.t. $x_1 - x_2 = 0$

$x^* = (3,3)$
Lagrange’s key insight

\[
\min_x \quad f(x) \\
g(x) = 0
\]
Lagrange’s key insight

V1: A statement on the gradient

\[ \nabla_x f(x) \bigg|_{x=x^*} = \lambda \nabla_x g(x) \bigg|_{x=x^*} \]
Lagrange’s key insight

V1: A statement on the gradient

\[ \max \min (f(x) - \lambda^T g(x)) \]

V2: A saddle point
Lagrange’s key insight

V1: A statement on the gradient

V2: A saddle point

V3: A game

(We will adopt this view)
A general theme in optimization is that it can be more efficient to phrase a problem as a saddle-point-finding exercise rather than as a difficult, pure optimization.
Dual Game: We control lambdas!

\[
\min_{x} \max_{\lambda} f(x) - \lambda^{T} g(x)
\]
Let’s play this game!

\[
\min_{x,y} \quad \frac{1}{2} (x^2 + y^2)
\]

\[
x - 1 = 0
\]

\[
y - 1 = 0
\]
Dual player is too aggressive ...
Stably change $\lambda$

Follow the Regularized Leader!

Specific FTRL: Gradient Descent
Augmented Lagrangian

\[
\min_x \; f(x) \\
\text{s.t.} \; g(x) = 0
\]

For \( t = 1 \ldots T \)

\begin{align*}
\text{Update} \; & \; x_t \\
& \quad x_{t+1} = \arg\min_x f(x) - \lambda_t^T g(x) + \eta g(x)^2 \\
\text{Update} \; & \; \lambda_t \\
& \quad \lambda_{t+1} = \lambda_t - \eta g(x_t)
\end{align*}

(Augmentation)
Augmented Lagrangian

\[
\min_x \max_{\lambda} f(x) - \lambda^T g(x)
\]

For \( t = 1 \ldots T \)

1. Update \( x_t \)
   \[
   x_{t+1} = \arg \min_x \left( f(x) - \lambda_t^T g(x) + \eta g(x)^2 \right)
   \]

2. Update \( \lambda_t \)
   \[
   \lambda_{t+1} = \lambda_t - \eta g(x_t)
   \]