Notes in Dynamics and Manuevering
ECEN 5008

Zachary Vogel

November 3, 2015

1 stuff for the day

Rough idea is that we want what is known as trajectory exploration.
Might ask: CAn our system $\dot{x} = f(x)$ do something like $(x_d(\cdot), u_d(\cdot))$.
Example: Sliding car $(v, \beta, \omega), (u_1, u_2 = \dot{\omega})$.
choose $x_d(\cdot), u_d(\cdot)$ based on $v_d(\cdot), a_{\text{lat},d}(\cdot)$.

$$v_d(t) = v_0$$

$$a_{\text{lat},d} = \text{a cubic looking function with flatness on both end}$$

$$\min \frac{||(x(\cdot), u(\cdot)) - (x_d(\cdot), u_d(\cdot))||_{L2}^2}{2}$$

such that $\dot{x} = f(x, u), x(0) = x_0$.
cost functional:

$$\int_0^T \frac{||x(\tau) - x_d(\tau)||_Q^2}{2} + \frac{||u(\tau) - u_d(\tau)||_R^2}{2} d\tau + \frac{||x(T) - x_d(T)||_{P1}^2}{2}$$

start with this because it is simpler than the best way, but a good place to start.
Playing around with PRONTO (Projection Operator Newton Trajectory Optimization).
Projection Operator means we will exploit a trajectory tracking controller.
Newton's method means we are hoping for a second order convergence.
Trajectory optimization because we want to search over curves x, u for best trajectory.

cost function $h(\xi) = \int_0^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T))$ where $\epsilon = (x(\cdot), u(\cdot))$.
Dynamics $\dot{x} = f(x, u, t)$. Won't always have a t, but might. $x(0) = x_0$.

$$u = \mu(t) + K(t)(\alpha(t) - x)$$

curve to try tracking $\mathfrak{P} : \xi = (\alpha(\cdot), \mu(\cdot)) \to \eta = (x(\cdot), u(\cdot))$

Properties:

1. $\forall \xi \in \text{domain}(\mathfrak{P}), \eta = \mathfrak{P}(\xi) \in \mathfrak{T}$
2. $\xi \in \mathfrak{T} \leftrightarrow \xi = \mathfrak{P}(xi)$
3. $(P)(\xi) = (P)((P)(\xi)) \forall \xi \in \text{domain}\mathfrak{P} : \mathfrak{P}^2 = \mathfrak{P}$i.e. it is a projection

The idea is that if you have a trajectory, you can push it a little bit to make the whole trajectory move.
Theorem for(trajecotry representation):
given $\xi \in \mathfrak{T}$, every nearby trajectory is of the form:

$$\eta = \mathfrak{P}(\xi + \zeta)$$

where $\zeta \in T_\xi \mathfrak{T}$is uniquely determined and where T is the tangent line space of ξ and $\mathfrak{T}$ is the trajectory.

$$\min_{\xi \in \mathfrak{T}} h(\xi)$$

equality constrained minimization problem.
our mapping $\mathfrak{P}$ is a mapping from a curve to a trajectory.

$$h(\mathfrak{P}(\xi)) =: g(\xi)$$

this is a cost of a trajectory generated from a curve.

$$\min_{\xi \in (\text{open set})} (g(\xi))$$

this is an unconstrained problem.
curves that project to a point with $\mathfrak{P}$ are called fibers.

The two problems are essentially equivalent, but one is constrained in the sense that the give the same ξ.
$\xi^*_{\text{c=constrained}}$ is a local min of unconstrained.
$\mathfrak{P}(\xi^*_{\text{u=unconstrained}}$ is a constrained local minimizer.

UNCONSTRAINED DESCENT directions.

$$f"\mathbb{R}^n \to \mathbb{R}$$

"steepest descent" or "gradient descent"

$$-\nabla f(x) = arg(\min_z(\partial f(x) * z + \frac{1}{2}||z||^2))$$

but gradient doesn't work in infiinite dimensions, so we use the left
Reece representation theorem in a hilbert space.

$$= arg(\min_z(< \nabla f(x), z > + \frac{1}{2} < z, z >))$$

here we are minimizing a quadratic model function.
prove the minimizer for that equation is infact the negative gradient.

Now we do it with Newton's Method:

$$-H(x)^{-1}\nabla f(x) = arg(\min_z(Df(x) * z + \frac{1}{2}D^2 f(x) * < z, z >)) = arg(\min_z(< \nabla f(x), z > + \frac{1}{2} < z, H(x)z >))$$

where H is the hessian.

$$h(\mathfrak{P}(\xi))$$

2 aside

$$\min f(x)$$

such that $g(x) = 0$

$$L(x, \lambda) = f(x) + \lambda^T g(x)$$

$$L_\lambda = 0, g(x) = 0$$

$$L_x = 0, \nabla f(x) + \sum \lambda_k \nabla g_k(x) = 0$$

$$f : \mathbb{R}^n \to \mathbb{R}$$

$$g : \mathbb{R}^m \to \mathbb{R}^m$$

$$\lambda \in (\mathbb{R}^m)^*$$

continuous linear functionals on $\mathbb{R}^m$.