WEC Systems Framework

Rebecca McCabe

November 2023

Contents

1 Control Co-Design Framework

I am trying to solve a control co-design problem for wave energy. This means simultaneously optimizing the physical device design and the controller behavior. The design variables, x, can be split into several categories. θ refers to decision variables related to the static (constant in time) WEC design. This includes θ_{lim} the limit variables (ie max force, position, power, and velocity), θ_{pto} the variables defining the PTO impedance (generator type, generator resistance/inductance, gear ratio, drivetrain inertia/stiffness), and θ_{mech} any remaining variables, mainly mechanical (ie hydrodynamic architecture, geometric dimensions, structural thicknesses, material choice). Meanwhile, x_2 and u are the fourier coefficients for the WEC velocity and powertrain force respectively, which are the decision variables related to the WEC dynamic response.

$$x = \begin{bmatrix} \theta_{lim} \\ \theta_{pto} \\ \theta_{mech} \\ x_2 \\ u \end{bmatrix} \tag{1}$$

The objective is broken up into J_{static} and $J_{dynamic}$. If the total objective J_{total} were to minimize LCOE for example, then the dynamic objective would be electrical power and the static objective would be cost. I am optimizing NVOE and net eco-value instead of LCOE, so it will be slightly different, but let's assume LCOE for the sake of understanding the problem structure.

The objectives have the following dependence on the design variables:

$$J_{total} = J_t(J_{static}, J_{dynamic}) \tag{2}$$

$$J_{static} = J_s(\theta_{lim}, \theta_{pto}, \theta_{mech}) \tag{3}$$

$$J_{dynamic} = J_d(\theta_{pto}, \hat{x}_2, \hat{u}) \tag{4}$$

1.1 Different CCD Structures

In the paper "Towards a Fair Comparison between the Nested and Simultaneous Control Co-Design Methods using an Active Suspension Case Study," Sundarrajan and Herber describe two possible strategies for control co-design and their pros and cons. In the nested strategy, the control is solved as an inner suboptimization within each iteration of the outer design optimization. In the simultaneous strategy, the control and design problem are solved in a single larger optimization.

Nested makes sense when the control sub-problem has appealing properties like convexity that enable a fast solution, which would not be possible if the problems were solved simultaneously because of nonconvexities in the design problem. A downside of nested is that the control subproblem can be infeasible for some designs, and the subproblem can interfere with derivatives. In the case study in the aforementioned paper, simultaneous was preferable when analytic gradients were available.

Question: is the outer nonlinear design problem in that paper convex? Why are they fixating on linear vs nonlinear instead of convex vs nonconvex?

Question: in my problem, are there any plant designs where the control problem would be infeasible? Maybe if the limits are too low.

1.2 Option 1: Nested

Inner (WecOptTool):

$$\begin{aligned} \min_{x_2,u} \quad & J_t(J_s(\theta_{lim},\theta_{pto},\theta_{mech}), J_d(\theta_{pto},x_2,u)) \\ \text{s.t.} \quad & h(\theta_{mech},x_2,u)=0 && \text{(dynamics)} \\ & g_1(\theta_{lim},x_2,u)\le 0 && \text{(control limits)} \end{aligned} \tag{5}$$

Outer:

$$\begin{aligned} \min_{\theta_{lim},\theta_{pto},\theta_{mech}} \quad & J_t(J_s(\theta_{lim},\theta_{pto},\theta_{mech}), \\ & \quad J_d(\theta_{pto},x_2^*(\theta_{lim},\theta_{pto},\theta_{mech}),u^*(\theta_{lim},\theta_{pto},\theta_{mech})) \\ \text{s.t.} \quad & g_2(\theta_{lim},\theta_{pto},\theta_{mech})\le 0 \quad \text{(design constraints)} \end{aligned} \tag{6}$$

the $*$ refers to the optimal value that results from performing the inner optimization.

1.2.1 Option 1a

Same as above but the objective for the inner optimization is just J_d instead of J_t. These are essentially equivalent when optimizing LCOE, but could be different when optimizing net value. This could potentially create convexity for the inner problem, but would result in a suboptimal solution if J_t is not monotonic wrt J_d.

1.3 Option 2: Simultaneous

$$\begin{aligned} \min_{\theta_{lim},\theta_{pto},\theta_{mech},x_2,u} \quad & J_t(J_s(\theta_{lim},\theta_{pto},\theta_{mech}), J_d(\theta_{pto},x_2,u)) \\ \text{s.t.} \quad & h(\theta_{mech},x_2,u)=0 && \text{(dynamics)} \\ & g_1(\theta_{lim},x_2,u)\le 0 && \text{(control limits)} \\ & g_2(\theta_{lim},\theta_{pto},\theta_{mech})\le 0 && \text{(design constraints)} \end{aligned} \tag{7}$$

1.4 Conclusion

It is not clear whether to go with 1, 1a, or 2. It could really depend on the convexity of each problem. It may be necessary to try both/all methods (for a smaller subset of the problem) and compare empirically to see which one takes less time and achieves the most optimal solution. Still, let's analyze convexity to understand if we even expect any of these problems to be convex.

2 WecOptTool Convexity

2.1 Motivation

It is highly desirable for optimization problems to be convex. In particular, if the WecOptTool problem can be made convex, it motivates that WEC DECIDER perhaps should be a nested optimization structure (option 1 above), where the inner problem is convex and the outer problem is nonconvex. With a convex inner problem, we can automatically get sensitivities, which act as derivatives to aid the convergence of the outer problem. It also guarantees a global optimum instead of a local optimum. It also is generally faster to solve.

2.2 Requirements for Convexity

There are three requirements to guarantee convexity in a disciplined convex programming framework:

1. The objective must either minimize a convex function or maximize a concave function.
2. The equality constraints must be affine = affine.
3. The inequality constraints must be convex $\leq$ concave and/or concave $\geq$ convex.

"Note that any constant expression is also affine, and any affine expression is convex and concave" `https://www.cvxpy.org/tutorial/dcp/index.html#dcp-problems`

2.3 Complex Variables

The design variables x_2 and u are real-valued and represent the cos and sin (real and imaginary) Fourier coefficients of the complex velocity and powertrain force. In WecOptTool, the definition is as follows, for any generic set of real-valued coefficients z:

$$z = \begin{bmatrix} Z_0 \\ \mathbb{R}\mathrm{e}(Z_1) \\ \mathbb{I}\mathrm{m}(Z_1) \\ \vdots \\ \mathbb{R}\mathrm{e}(Z_{N_{freq}-1}) \\ \mathbb{I}\mathrm{m}(Z_{N_{freq}-1}) \\ \mathbb{R}\mathrm{e}(Z_{N_{freq}}) \end{bmatrix} \tag{8}$$

so z has length $2N_{freq}$ because the first element (DC gain) and last element (highest frequency) have zero imaginary part.

In the equations in the following sections, instead of the real-valued coefficients z (the design variables x_2 and u), we need to use the complex values $\hat{z}$ ($\hat{x}_2$ and $\hat{u}$), defined as follows:

$$\hat{z} = \begin{bmatrix} Z_0 \\ Z_1 \\ \vdots \\ Z_{N_{freq}-1} \\ Z_{N_{freq}} \end{bmatrix} = \begin{bmatrix} Z_0 \\ \mathbb{R}\mathrm{e}(Z_1) + 1j\,\mathbb{I}\mathrm{m}(Z_1) \\ \vdots \\ \mathbb{R}\mathrm{e}(Z_{N_{freq}-1}) + 1j\,\mathbb{I}\mathrm{m}(Z_{N_{freq}-1}) \\ \mathbb{R}\mathrm{e}(Z_{N_{freq}}) \end{bmatrix} \tag{9}$$

where $1j$ is the imaginary unit and $\hat{z}$ has length $N_{freq} + 1$.

The following matrix equation is used to transfer between z and $\hat{z}$:

$$\hat{z} = Az \text{ where } A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 1 & 1j & 0 & 0 & & 0 \\ 0 & 0 & 0 & 1 & 1j & \dots & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & 0 & 0 & & 1 \end{bmatrix} \tag{10}$$

and A has size $N_{freq} + 1$ x $2N_{freq}$.

2.4 Guaranteed Convexity in Giorgio's Paper

2.4.1 Objective

In Giorgio's paper Numerical Optimal Control of Wave Energy Converters, he assumes linear dynamics. This means the WEC velocity $\hat{x}_2$ is a linear function of the control and excitation $\hat{u}$ and $\hat{e}$ (eqn 36 from that paper):

$$G\hat{x}_2 = \hat{u} + \hat{e} \tag{11}$$

This can be substituted into the mechanical power objective (eqn 26) $J^N = -\frac{T}{2}\hat{u}^T\hat{x}_2$ to get (eqn 39)

$$J^N = -\frac{T}{2}\hat{u}^T G^{-1}\hat{u} - \frac{T}{2}\hat{u}^T G^{-1}\hat{e} \tag{12}$$

This objective is convex with respect to the decision variables $\hat{u}$ because all diagonal elements of G are positive (the radiation damping). See his paper for further details. This meets requirement 1.

2.4.2 Constraints

Because the dynamics are incorporated into the objective by construction, there is no need for any equality constraints. This meets requirement 2.

The inequality constraints applied are limits on the position and force (eqns 47 and 49):

$$\begin{gathered}\begin{bmatrix}\Theta \\ -\Theta\end{bmatrix} \hat{u} \leq 1_{2(N_c+1)x1} F_{max} \\ \begin{bmatrix}\Theta \\ -\Theta\end{bmatrix} D_\phi^{-1} G^{-1} \hat{u} \leq 1_{2(N_k+1)x1} Z_{max} - \begin{bmatrix}\Theta \\ -\Theta\end{bmatrix} D_\phi^{-1} G^{-1} \hat{e}\end{gathered} \tag{13}$$

which are both linear (affine) with respect to the decision variable $\hat{u}$, which satisfies requirement 3.

Not only is this problem convex, it is a convex quadratic program, which is even easier to solve than a general convex problem.

2.5 Nonconvexity in WecOptTool

2.5.1 Objective

The objective in WecOptTool is electrical power, rather than mechanical power, which makes the objective more complicated. From the WecOptTool IEEE CCD paper eq 29 and 11, with variables renamed to match the conventions used above:

$$\begin{gathered}\bar{P}_{elec} = \frac{1}{2} \mathbb{R}\mathrm{e} \left(\hat{I}^* \hat{V} \right) \\ \begin{bmatrix}\hat{I} \\ \hat{V}\end{bmatrix} = Z_{abcd} \begin{bmatrix}\hat{x}_2 \\ \hat{u}\end{bmatrix}\end{gathered} \tag{14}$$

Plugging in:

$$\begin{aligned}\bar{P}_{elec} &= \frac{1}{2} \mathbb{R}\mathrm{e} \left(\begin{bmatrix}\hat{x}_2 \\ \hat{u}\end{bmatrix}^* Z_{abcd}^* \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix} Z_{abcd} \begin{bmatrix}\hat{x}_2 \\ \hat{u}\end{bmatrix} \right) \\ &= \frac{1}{2} \mathbb{R}\mathrm{e} \left(\begin{bmatrix}x_2 \\ u\end{bmatrix}^T A^* Z_{abcd}^* \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix} Z_{abcd} A \begin{bmatrix}x_2 \\ u\end{bmatrix} \right)\end{aligned} \tag{15}$$

So we see that the objective is quadratic, but asymmetric. Using the property $x^T M x = x^T \frac{M+M^T}{2} x \ \forall x, M$, we equivalently write the symmetricalized version:

$$\bar{P}_{elec} = \frac{1}{4} \mathbb{R}\mathrm{e} \left(\begin{bmatrix}x_2 \\ u\end{bmatrix}^T A^* Z_{abcd}^* \begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix} Z_{abcd} A \begin{bmatrix}x_2 \\ u\end{bmatrix} \right) \tag{16}$$

This can be rewritten as a quadratic problem:

$$\bar{P}_{elec} = \begin{bmatrix}x_2 \\ u\end{bmatrix}^T Q \begin{bmatrix}x_2 \\ u\end{bmatrix} \text{ where } Q = \frac{1}{4} \mathbb{R}\mathrm{e} \left(A^* Z_{abcd}^* \begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix} Z_{abcd} A \right) \tag{17}$$

Unfortunately, the eigenvalues of $\begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix}$ are -1 and 1, so the matrix Q is neither positive semidefinite nor negative semidefinite, and the objective is not convex.

2.5.2 Constraints

WecOptTool does not assume linear dynamics, so we don't solve for $\hat{x}_2$ as a linear function of the control and excitation. Instead we enforce the dynamics as an equality constraint (eqn 4.96 of Giorgio's thesis, which I have split up to better match the organization of WecOptTool source code):

$$I_y \Phi_j D_\phi \hat{x}_2 + F_{hydro}(t_j) + F_{add}(t_j) = 0 \tag{18}$$

This time domain equation states that the mass/inertia times acceleration (ma) equals the sum of forces. This equation is enforced separately at each timestep t_j, so it is actually $N_{timesteps}$ separate equality constraints.

F_{hydro} encompasses friction, radiation, hydrostatic, and excitation (Froude-Krylov and diffraction) forces, which are all linear:

$$F_{hydro}(t_j) = B_{v_1} \Phi_j \hat{x}_2 + \Phi_j G \hat{x}_2 + S_h \Phi_j \hat{x}_1 - \gamma_e(t_j) \tag{19}$$

and F_{add} encompasses the powertrain force (linear) plus any user-defined forces (potentially nonlinear). In Giorgio's thesis, quadratic drag is used as the nonlinear user-defined force:

$$F_{add}(t_j) = -\Phi_j \hat{u} + B_{v_2} \Phi_k \hat{x}_2 |\Phi_k \hat{x}_2| \tag{20}$$

This equality constraint only meets requirement 2 if there are no nonlinear terms in F_{add}.

Inequality constraints are up to the user to add. Typical constraints will be force and position limits, which are linear the same way they were in the previous section, along with peak power limits and a generator force limit which is only active above certain speeds, which are quadratic and linear respectively, the latter requiring the addition of a binary indicator decision variable for each timestep, which makes it a mixed integer problem. The force, position, and conditional force limit are convex (affine), but the power limit is not (nonconvex quadratic).

In summary, the general WecOptTool problem has a nonconvex quadratic objective, nonconvex quadratic inequality constraints, and potentially nonlinear equality constraints. It meets none of the three requirements for convexity. If no nonlinear force terms are used, it could be solved as a nonconvex MI-QCQP (mixed integer quadratically constrained quadratic program), but otherwise must be solved as a general NLP. The current WecOptTool uses the SLSQP NLP algorithm from scipy.optimize.

2.6 Convexifying WecOptTool

I would like to explore how to convexity this problem for the reasons mentioned in the motivation section.

2.6.1 Equality Constraint

Let's start with the dynamics equality constraint in the case where a nonlinear F_{add} term is used. Recall that for convexity, equality constraints must be affine in the decision variables. I can think of two ways to do this:

1. Replace nonlinear dynamics terms with Fourier coefficients (decision variables) and add convex inequality constraints to those coefficients

2. Replace nonlinear dynamics terms with Fourier coefficients (constants) and add nonconvex constraints to those coefficients in the outer problem

Item 1 would only be possible when the F_{add} terms are still convex even though they are not affine, and where an inequality constraint would be tight and therefore equivalent to the equality constraint. An example of this is drag. Drag is quadratic in the velocity x_2, and because drag involves power dissipation it is reasonable to assume that an inequality constraint of the form $F_{drag} \geq x_2^T C x_2$ would be tight because drag should always decrease the power generation. (Need to check this assumption more rigorously, what if there is some case where the drag is helpful because it helps stay under some max speed constraint?) The downside of this would be that we have added more design variables to the inner problem, in the form of the Fourier coefficients of drag $\hat{F}_{drag}$, but it will likely still be more computationally efficient to solve a large convex problem than a small nonconvex one.

Question: is this equivalent to just relaxing the equality constraint to an inequality? Then I wouldn't need to add design variables. I think it's not equivalent though.

Item 2 is the alternative for when the F_{add} cannot be written convexly or would not be tight as an inequality. This would blur the lines between the inner and outer problems, essentially making some of the dynamics a part of the outer problem. The downside here is that now we have added more design variables to the outer problem, which is worse than doing so for the inner problem because the outer problem is nonconvex. It is unclear whether this strategy would even be beneficial, compared to keeping both problems nonconvex but small (without any extra decision variables).

The main nonlinear F_{add} relevant to wave energy are drag, mooring forces, and second order hydrodynamic forces. Drag can be handled with the first option, and I don't know enough about the form of the other two forces to know whether they can, but it seems plausible that they can at least be approximated convexly, so for now I will assume it can be done. With the equality constraints now effectively affine, I will move onto the objective and inequalities.

2.6.2 Objective

At this point with the equality affine, we officially have a nonconvex MI-QCQP. This can be solved as-is and would be more efficient than NLP, but I'm not sure whether it would have the global optimality and especially the sensitivities that are possible to get with a convex program. So, here are four ideas to make it convex:

1. Add more constraints so that the problem is convex in the entire feasible region
2. Assume positive fourier coefficients variables and use SOCP
3. Log-log transform (Geometric programming)
4. Solve the now-linear equality constraint and plug it into the objective to make the QP convex, like it was in Giorgio's paper

Item 1's concept is based on work by Zhong and Yeung 2018 (OMAE) that shows that constraining the slew rate of PTO force can guarantee convexity for a single WEC, and Zhong and Yeung 2022 (Ocean Engineering) shows that constraining reactive power to zero (pure damping control) can guarantee convexity for an array. Constraining PTO force slew rate is reasonable because the low-level controllers will have finite bandwidth, but constraining reactive power is rather limiting. Their formulation is model predictive control rather than pseudo-spectral method, so I'm not sure if these constraints would also make my problem convex. I have not looked into it extensively. It seems like CVXPy would not count this as convex, even if it is, because CVXPy looks for convexity over the whole domain, not just the feasible region. I could veryify it by hand and then plug directly into a solver without CVXPy, but CVXPy is the tool that provides the sensitivities that I want, so this is not ideal. Maybe there are other tools out there that provide sensitivities of QPs.

Item 2 - I have implemented it (see code here) and gotten it to work on a toy problem that maximizes mechanical power only for a single frequency. Rather than maximize power, I maximize the square root of power, using CVXPy's geo_mean function. It is nonlinear, but CVXPy recognizes it as convex and solves it as a second order cone program SOCP. This requires that all Fourier coefficients are positive. If the design variables were $\hat{I}$ and $\hat{V}$ directly that would be true, but because they are the mechanical variables rather than the electrical variables, I'm not sure if that's an appropriate assumption. I think the criterion would be that Z_{abcd} needs to be positive semidefinite for it to be true. From the WecOptTool CCD paper:

$$Z_{abcd} = \begin{bmatrix} -Z_{FI}^{-1} Z_{FU} & Z_{FI}^{-1} \\ Z_{VU} - Z_{VI} Z_{FI}^{-1} Z_{FU} & Z_{VI} Z_{FI}^{-1} \end{bmatrix} \tag{21}$$

Z_{FI} and Z_{VU} are real negative values. Z_{FU} has a negative real part and an imaginary part that can be of either sign. Z_{VI} has a positive real part and a positive imaginary part. I am not sure how to tell from these signs whether Z_{abcd} is positive semidefinite.

If Z_{abcd} isn't positive semidefinite, I suppose there would always be the possibility of changing the design variables to $\hat{I}$ and $\hat{V}$, and then using Z_{abcd} to rewrite the constraints in terms of $\hat{I}$ and $\hat{V}$.

The only downside of this idea is that it's a SOCP instead of a QP, but I don't think this is a very big deal.

Edit: actually, this might not extend to multiple frequencies (my toy problem was only a single frequency). Because $\sqrt{P_1 + P_2} \neq \sqrt{P_1} + \sqrt{P_2}$, ie the powers from different frequencies do not sum correctly in this case. Need to investigate, could be a dealbreaker.

Item 3 seems plausible at first, because power is a posynomial (a sum of positive products), but the problem here comes from the constraints. Only posynomial inequality constraints are allowed, but the velocity-conditioned force limit is polynomial (it has a negative term). This constraint could possibly be reformulated but I'm not sure. Plus, only monomial equality constraints are allowed, but I have a posynomial (or perhaps polynomial) equality constraint. This constraint would have to be relaxed to an inequality constraint, and if there are any nonpositive terms it's pretty much game over.

Item 4 is appealing because if the substitution made the matrix positive definite, it would make the objective a convex quadratic program, which is even easier to solve than the convex SOCP which we got in item 2. We know from Giorgio's paper that it is convex in the case of maximizing mechanical power, but I am not sure if this is still true when we are maximizing mechanical power. It seems like Z_{abcd} being positive semidefinite would also be the criterion here. A downside of this approach would be more implementation effort, because it would require solving a matrix equation for the dynamics where previously there was an equality constraint.

In summary, item 1 would yield a non-CVXPy-compatible convex QP and requires more investigation to see if it is possible. Item 2 yields a convex SOCP and is possible if either Z_{abcd} is positive semidefinite or the design variables are changed (edit: and if I confirm whether it extends to multiple frequencies!). Item 3 would yield a convex GP and requires relaxing the dynamics to an inequality and has potential sign issues in the constraints. Item 4 would yield a convex QP and requires Z_{abcd} positive semidefinite.

2.6.3 Inequality Constraint

The only inequality constraint of issue is the peak power constraint, which is quadratic. Potentially some of the same tricks for the objective could be employed, as described in the previous section. Otherwise this constraint could simply be approximated as linear, or even as a series of conditionally-active linear constraints to get a bit tighter, which seems acceptable given that real torque-speed curves don't perfectly limit mechanical power anyway, especially when there is no field weakening (which is already assumed given the linear generator impedance model). I have no other ideas.

2.7 Next steps

- Check whether maximizing the square root of power (or simliar SOCP) extends to multiple frequencies

- Check whether Z_{abcd} is positive semi-definite
- Confirm whether drag constraint would be tight and whether making drag an inequality is equivalent relaxing the entire dynamics constraint
- Check whether there are other ways besides CVXPy to get sensitivities, for the convex-only-where-feasible MIQP
- Confirm that a nonconvex QP is more difficult to solve than a convex SOCP with potentially larger size. Can nonconvex QP give a global optimum or sensitivities?
- Check whether GP constraints are valid (negatives)
- Investigate convexity if I used optimize for net value instead of LCOE
- See if the outer problem was convex in Dan Herber's CCD case study
- Are there any plant designs where the control problem is infeasible?
- Think about whether mooring and second order hydro forces are convex/tight like drag probably is
- Look into which of these problems are DPP (differential parametric programming) compliant, which is a requirement for CVXPy to provide the sensitivities

2.8 Implementation

Solvers that are accessible from CVXPy and can solve MISOCPs (mixed-integer second order cone programs) and MIQPs (mixed-integer quadratic programs):

- SCIP (open source, what I'm using now)
- ECOS-BB (open source, not recommended due to correctness issues)
- CPLEX (commercial, free license for academia)
- GUROBI (commercial, free license for academia)
- MOSEK (commercial, free license for academia)
- XPRESS (commercial, free community edition if variables + constraints ¡ 5000)