To avoid asymptotic behavior of the denominator polynomial, we must constrain the non-linear solver.

Suppose we have a 3rd order polynomial in the denominator (note that the coefficients are complex),

The complex roots of the polynomial must be constrained such that

Where s is the space of all complex inputs fed into the predistorter, and rk is the kth root.

One method of applying this constraint is by using MATLAB’s fmincon( ), which allows a nonlinear inequality to be specified in the form:

Considering our requirement from Eqn (1), the nonlinear expression ‘c’ will be of the form:

Where is a user-specified tolerance margin between the maximum magnitude of the input signal, and the magnitude of the complex root rk.

How should rk be determined for a variable order polynomial?

Option 1: for each order of polynomial, write a non-linear expression for the roots rk in terms of the coefficients (a,b,c,d). This will only work for polynomials of 4th order or lower because analytic closed form expressions for higher order polynomials do not exist (Abel-Ruffini theorem). Also, expressions for the roots of cubic and quartic polynomials (Cardano and Ferrari’s solutions) do not yield correct results when the coefficients are complex (tested in MATLAB).

Option 2: Use MATLAB’s roots function to determine the roots |rk|. I am not sure how fmincon( ) will behave when the non-linear expression ‘c’ is instead replaced by a function.