A typical design process for real-time embedded systems involves choosing the values of certain system parameters, followed by a timing/performance analysis to determine whether all design constraints have been satisfied. Such parameters might range from activation rates of different tasks and task deadlines, to task partitioning and mapping decisions. If some of the design constraints fail to get satisfied, one or more system parameters are changed and the analysis is invoked once again. This iteration is repeated in an interactive fashion till a satisfactory design is obtained.

However, it turns out that for most real-life systems, timing/performance analysis is computationally expensive and hence such an interactive performance debugging cycle is very tedious. Further, such designs often involve multiple performance trade-offs and computing the underlying trade-off or pareto-curve is also a computationally expensive, and often infeasible, procedure.

In this talk I shall describe some of our recent efforts in addressing these problems. In particular, I will outline an "interactive schedulability analysis" algorithm that exploits the fact that with small changes in the system parameters, the full schedulability analysis machinery need not be invoked. This easily leads to more than 20x speedups in the analysis and greatly simplifies the above-mentioned interactive performance debugging cycle. I will also describe a related algorithm that efficiently, but approximately, computes trade-off curves and as a result meaningfully exposes the different performance trade-offs involved in a design.