

Algebraic Invariant Theory in Systems Research

Lecture 2

Examples, Applications, and Future Research

Rudolf Kalman¹

ABSTRACT

Since the data in the Foster-Ladenheim catalog are more than seventy-five (75) years old (see first lecture), some of you may regard talking about it as scientific archeology. But no, our forefathers, the old boys, surprisingly, knew much more about things than we thought they did.

Actually, the Ladenheim catalog is a rich source of information, some on the surface, some to be made precise.

Foster's contributions:

(1) The idea of a "generic" network. ("Generic" means: no further simplification possible without affecting basic properties; nowhere explicitly defined by Foster, even though the celebrated "Foster and Cauer canonical forms", discovered in the 1920's (see Guillemin's book) are, of course, generic networks and duly appear in the catalog.)

(2) That straightforward algebraic manipulation can lead to explicit solution of the realization problem. A rediscovery of some tricks of classical invariant theory.

Foster's omissions:

(3) Calling "discriminant" the "resultant" in the 1962 survey paper, in spite of accurate use of the traditional Sylvester definition of the resultant. (Cultural obstacles between engineering and mathematics, but also sloppy teaching of mathematics to engineering students in the Boston area in the early twentieth century.)

(4) Not recognizing that the network realization problem, as reformulated by himself, shows many affinities (use of resultants and related invariants) to the classical theory of projective invariants involving a pair of polynomials, well developed before 1900.

(5) Getting actual criteria for the positive realizability of a fixed impedance. (Very difficult problem, still largely unsolved; perhaps already solved, for a large class of networks, by "regularity" in the sense of Malcolm Smith and Jason Jiang, ...)

Simple examples will help to tie some of these things further together.

Who knows what the future may hold? After all, the genetic code looks like a realization problem.

¹ kalman@math.ethz.ch