computers, databases, and instruments.) A user of yourSkyG can specify parameters describing a mosaic to be constructed. yourSkyG then constructs the mosaic on the IPG and makes it available for downloading by the user. The complexities of determining which input images are required to construct a mosaic, retrieving the required input images from remote sky-survey archives, uploading the images to the computers on the IPG, performing the computations remotely on the Grid, and downloading the resulting mosaic from the Grid are all transparent to the user.

This program was written by Joseph Jacob, James Collier, Loring Craymer, and David Curkendall of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827.Refer to NPO-40761.

ANTLR Tree Grammar Generator and Extensions

A computer program implements two extensions of ANTLR (Another Tool for Language Recognition), which is a set of software tools for translating source codes between different computing languages. ANTLR supports predicated-LL(k) lexer and parser grammars, a notation for annotating parser grammars to direct tree construction, and predicated tree grammars. ["LL(k)" signifies "left-right, leftmost derivation with k tokens of look-ahead," referring to certain characteristics of a grammar.] One of the extensions is a syntax for tree transformations. The other extension is the generation of tree grammars from annotated parser or input tree grammars. These extensions can simplify the process of generating source-to-source language translators and they make possible an approach, called "polyphase parsing," to translation between computing languages. The typical approach to translator development is to identify high-level semantic constructs such as "expressions," "declarations," and "definitions" as fundamental building blocks in the grammar specification used for language recognition. The polyphase approach is to lump ambiguous syntactic constructs during parsing and then disambiguate the alternatives in subsequent tree transformation passes. Polyphase parsing is believed to be useful for generating efficient recognizers for C++ and other languages that, like C++, have significant ambiguities.

This program was written by Loring Craymer of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-30565.

Generic Kalman Filter Software

The Generic Kalman Filter (GKF) software provides a standard basis for the development of application-specific Kalman-filter programs. Historically, Kalman filters have been implemented by customized programs that must be written, coded, and debugged anew for each unique application, then tested and tuned with simulated or actual measurement data. Total development times for typical Kalman-filter application programs have ranged from months to weeks. The GKF software can simplify the development process and reduce the development time by eliminating the need to re-create the fundamental implementation of the Kalman filter for each new application. The GKF software is written in the ANSI C programming language. It contains a generic Kalman-filter-development directory that, in turn, contains a code for a generic Kalman filter function; more specifically, it contains a generically designed and generically coded implementation of linear, linearized, and extended Kalman filtering algorithms, including algorithms for state- and covariance-update and -propagation functions. The mathematical theory that underlies the algorithms is well known and has been reported extensively in the open technical literature. Also contained in the directory are a header file that defines generic Kalman-filter data structures and prototype functions and template versions of application-specific subfunction and calling "navigation/estimation" routine code and headers. Once the user has provided a calling routine and the required application-specific subfunctions, the application-specific Kalman-filter software can be compiled and executed immediately. During execution, the generic Kalman-filter function is called from a higher-level "navigation" or "estimation" routine that preprocesses measurement data and postprocesses output data. The generic Kalman-filter function uses the aforementioned data structures and five implementation-specific subfunctions, which have been developed by the user on the basis of the aforementioned templates. The GKF software can be used to develop many different types of unfactorized Kalman filters. A developer can choose to implement either a linearized or an extended Kalman filter algorithm, without having to modify the GKF software. Control dynamics can be taken into account or neglected in the filter-dynamics model. Filter programs developed by use of the GKF software can be made to propagate equations of motion for linear or nonlinear dynamical systems that are deterministic or stochastic. In addition, filter programs can be made to operate in user-selectable "covariance analysis" and "propagation-only" modes that are useful in design and development stages.

This program was written by Michael E. Lisano II and Edwin Z. Crues of LinCom for **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23006