What Is The Matlab Code For Adaptive Gamma Correction In

Undocumented Secrets of MATLAB-Java Programming
Optics Using MATLAB
Analysis of the MPEG-1 Layer III (MP3) Algorithm using MATLAB
MATLAB PROGRAMMING. SCRIPTS and FUNCTIONS
A Tool for the Generation of Naturally Parallel Matlab Code
A Full Matlab Code for Computing a Novel Hybrid Solc Filter
MATLAB APP Designer: Learn By Example (UUM Press)
MATLAB and Simulink Code Generation
MATLAB kompakt
Full Matlab Code for Synthesis and Optimization of Bragg Gratings
Radar Signal Analysis and Processing Using MATLAB
The Elements of MATLAB Style
MATLAB Programming for Biomedical Engineers and Scientists
Signal Processing in MATLAB. Simulink Blocks and Code Generation
Matlab Code Generation
Practical Optimization with MATLAB
MATLAB Software for the Code Excited Linear Prediction Algorithm
Matlab für Dummies
Numerical Methods using MATLAB
Circuit Analysis II
Applied Signal Processing
Code Generation in Simulink
Vibration Simulation Using MATLAB and ANSYS
Matlab Code for Sorted Real Schur Forms
Introduction to Intelligent Systems, Control, and Machine Learning using MATLAB
MATLAB Recipes
A Guide to MATLAB Object-Oriented Programming
Accelerating MATLAB Performance
The Dominant Pole Design Toolbox - the Matlab Code
Numerical Analysis Using MATLAB and Spreadsheets
A MATLAB Exercise Book
MATLAB Implementation of the Steganographic Algorithm F5
Robotics, Vision and Control
Optical Interference Filters Using Matlab
A Practical Guide to Error-Control Coding Using MATLAB
Exploratory Data Analysis with MATLAB
Radar Systems Analysis and Design Using MATLAB Second Edition
Diagnostic Radiology Physics with MATLAB®
Optimization of Matlab code using delayed evaluation and runtime code generation with the TaskGraph library

KENDAL MCCONNELL

Undocumented Secrets of MATLAB-Java Programming Artech House
You can use MATLAB and Simulink for rapid prototyping of hardware designs. Some DSP System Toolbox algorithms support HDL code generation when used with HDL Coder. For digital filter designs in MATLAB, use Filter Design HDL Coder to generate HDL code and optimize filter implementation for hardware speed or area. Both HDL Coder and Filter Design HDL Coder generate target-independent synthesizable Verilog and VHDL code for FPGA programming or ASIC prototyping and design. They also generate scripts and test benches for use with third-party HDL simulators.
Optics Using MATLAB CRC Press
This practical resource provides you with a comprehensive understanding of error control coding, an essential and widely applied area in modern digital communications. The goal of error control coding is to encode information in such a way that even if the channel (or storage medium) introduces errors, the receiver can correct the errors and recover the original transmitted information. This book includes the most useful modern and classic codes, including block, Reed Solomon, convolutional, turbo, and LDPC codes.You find clear guidance on code construction, decoding algorithms, and error correcting performances. Moreover, this unique book introduces computer simulations integrally to help you master key concepts. Including a companion DVD with MATLAB programs and supported with over 540 equations, this hands-on reference provides you with an in-depth treatment of a wide range of practical implementation issues.
Analysis of the MPEG-1 Layer III (MP3) Algorithm using MATLAB Orchard Publications
This book presents a theoretical description of fiber Bragg gratings, focusing on channels' densification and the tunability of Bragg filters. It also includes a full Matlab code for the synthesis and optimization of several kinds of fiber Bragg gratings by using the directed tabu search, the simulated annealing method and the genetic algorithm. Physical and optical parameters of uniform, chirped and sampled fiber Bragg gratings are then reconstructed with these algorithms.
MATLAB PROGRAMMING. SCRIPTS and FUNCTIONS Cambridge University Press
Der Autor bietet eine in die einzelnen Fachgebiete gruppierte Dokumentation der rund 1000 MATLAB-Befehle. Die Funktionen der einzelnen Befehle werden verständlich erläutert und anhand zahlreicher praxisorientierter Beispiele und Abbildungen verdeutlicht. Der umfangreiche Index und die klare Strukturierung vervollständigen das Buch und ermöglichen einen effizienten, praxisgerechten Einsatz des Buches und damit auch von MATLAB selbst. Die 6., akualisierte und erweiterte Auflage wurde an die aktuelle MATLAB-Version mit vielen Änderungen gegenüber der Vorgängerversion angepasst. Neue Datentypen, neue Funktionalitäten und Visualisierungen beruhen auf einem neuen Kernel, neue Funktionalitäten für grafische Benutzeroberflächen, deutlich erweiterte Möglichkeiten im Rahmen der Objekt-Orientierten Programmierung (OOP), Unit Testing.
A Tool for the Generation of Naturally

Parallel Matlab Code CRC Press
MATLAB App Designer is a feature that allows MATLAB code to be packaged into an interactive software. The software can be shared on any computer without the trouble of having to install MATLAB or even knowing programming knowledge to be able to operate the software. This books provides hands on approach to guide learners in developing the software from scratch using MATLAB App Designer. It covers a wide variety on standard graphical component (radio button, tables, button, check boxes, sliders and many others) and how to utilize its properties and function in deploying end user software. Source code for all the example program can be studied and understand by student easily. This equips learners with the fundamental and required skills for developing the application on their own. Added that, the example code can be reusable with other case problem or application similar to the hands on example. The key to mastering any application development software is to practice, so that you are familiarize with the components and understand its properties and behavior. In simple word, knowing how each components work is essential. This is where this book benefits learner that needs to develop software application using MATLAB.
SPIE-International Society for Optical Engineering
MATLAB Programming for Biomedical Engineers and Scientists provides an easy-to-learn introduction to the fundamentals of computer programming in MATLAB. This book explains the principles of good programming practice, while demonstrating how to write efficient and robust code that analyzes and visualizes biomedical data. Aimed at the biomedical engineer, biomedical scientist, and medical researcher with little or no computer programming experience, it is an excellent resource for learning the principles and practice of computer programming using MATLAB. This book enables the reader to: Analyze problems and apply structured design methods to produce elegant, efficient and well-structured program designs Implement a structured program design in MATLAB, making good use of incremental development approaches Write code that makes good use of MATLAB programming features, including control structures, functions and advanced data types Write MATLAB code to read in medical data from files and write data to files Write MATLAB code that is efficient and robust to errors in input data Write MATLAB code to analyze and visualize medical data, including imaging data Many real world biomedical problems and data show the practical application of programming concepts Two whole chapters dedicated to the practicalities of designing and implementing more complex programs An accompanying website containing freely available data and source code for the practical code examples, activities, and exercises in the book For instructors, there are extra teaching materials including a complete set of slides, notes for a course based on the book, and course work suggestions
A Full Matlab Code for Computing a Novel Hybrid Solc Filter Apress
Designed for use in a second course in circuit analysis, this text engages a full spectrum of circuit analysis related subjects ranging from the most abstract to the most practical. Featured are methods of expressing signals in terms of the elementary functions, an introduction to second order circuits, and several examples of analysing electric

circuits using Laplace transformation methods. Though not written explicitly to be used with MATLAB, this text provides many useful tips and strategies for MATLAB, allowing students to get the most out of the popular program. All of the information provided is designed to be covered in one semester or two quarters.

MATLAB APP Designer: Learn By Example (UUM Press) GRIN Verlag

This book describes several modules of the Code Excited Linear Prediction (CELP) algorithm. The authors use the Federal Standard-1016 CELP MATLAB(r) software to describe in detail several functions and parameter computations associated with analysis-by-synthesis linear prediction. The book begins with a description of the basics of linear prediction followed by an overview of the FS-1016 CELP algorithm. Subsequent chapters describe the various modules of the CELP algorithm in detail. In each chapter, an overall functional description of CELP modules is provided along with detailed illustrations of their MATLAB(r) implementation. Several code examples and plots are provided to highlight some of the key CELP concepts. Table of Contents: Introduction to Linear Predictive Coding / Autocorrelation Analysis and Linear Prediction / Line Spectral Frequency Computation / Spectral Distortion / The Codebook Search / The FS-1016 Decode

MATLAB and Simulink Code Generation Springer Science & Business Media

The practice of robotics and computer vision both involve the application of computational algorithms to data. Over the fairly recent history of the fields of robotics and computer vision a very large body of algorithms has been developed. However this body of knowledge is something of a barrier for anybody entering the field, or even looking to see if they want to enter the field — What is the right algorithm for a particular problem?, and importantly, How can I try it out without spending days coding and debugging it from the original research papers? The author has maintained two open-source MATLAB Toolboxes for more than 10 years: one for robotics and one for vision. The key strength of the Toolboxes provide a set of tools that allow the user to work with real problems, not trivial examples. For the student the book makes the algorithms accessible, the Toolbox code can be read to gain understanding, and the examples illustrate how it can be used —instant gratification in just a couple of lines of MATLAB code. The code can also be the starting point for new work, for researchers or students, by writing programs based on Toolbox functions, or modifying the Toolbox code itself. The purpose of this book is to expand on the tutorial material provided with the toolboxes, add many more examples, and to weave this into a narrative that covers robotics and computer vision separately and together. The author shows how complex problems can be decomposed and solved using just a few simple lines of code, and hopefully to inspire up and coming researchers. The topics covered are guided by the real problems observed over many years as a practitioner of both robotics and computer vision. It is written in a light but informative style, it is easy to read and absorb, and includes a lot of Matlab examples and figures. The book is a real walk through the fundamentals of robot kinematics, dynamics and joint level control, then camera models, image processing, feature extraction and epipolar geometry, and bring it all

together in a visual servo system. Additional material is provided at http://www.petercorke.com/RVC
MATLAB kompakt Morgan & Claypool Publishers
A Guide to MATLAB Object-Oriented Programming is the first book to deliver broad coverage of the documented and undocumented object-oriented features of MATLAB®. Unlike the typical approach of other resources, this guide explains why each feature is important, demonstrates how each feature is used, and promotes an understanding of the interactions between features. Assuming an intermediate level of MATLAB programming knowledge, the book not only concentrates on MATLAB coding techniques but also discusses topics critical to general software development. It introduces fundamentals first before integrating these concepts into example applications. In the first section, the book discusses eight basic functions: constructor, subsref, subsasgn, display, struct, fieldnames, get, and set. Building on the previous section, it explores inheritance topics and presents the Class Wizard, a powerful MATLAB class generation tool. The final section delves into advanced strategies, including containers, static variables, and function fronts. With more than 20 years of experience designing and implementing object-oriented software, the expert author has developed an accessible and comprehensive book that aids readers in creating effective object-oriented software using MATLAB.
Full Matlab Code for Synthesis and Optimization of Bragg Gratings Orchard Publications
Scripts are the simplest kind of program fil because they have no input or output arguments. They are useful for automating series of MATLAB commands, such as computations that you have to perform repeatedly from the command line or series of commands you have to reference.Scripts and functions allow you to reuse sequences of commands by storing them in program files Scripts are the simplest type of program, since they store commands exactly as you would type them at the command line. However, functions are more flexibland more easily extensible. instead of manually updating the script each time, you can make your programmore flexibl by converting it to a function. Replace the statements that assign values to band h with a function declaration statement. The declaration includes the function keyword, the names of input and output arguments, and the name of the function. Functions have their own workspace, separate from the base workspace.MATLAB scripts, including live scripts, can contain code to defin functions. These functions are called local functions. Local functions are useful if you want to reuse codewithin a script. By adding local functions, you can avoid creating and managing separate function files They are also useful for experimenting with functions, which can be added, modified and deleted easily as needed.Local functions are only visible within the fil where they are defined both to the script code and other local functions within the file They are not visible to functions in otherfiles and cannot be called from the command line. They are equivalent to subroutines inother programming languages, and are sometimes called subfunctions. MATLAB live scripts and live functions are interactive documents that combine MATLAB code with formatted text, equations, and images in a single environment called the LiveEditor. In

addition, live scripts store and display output along side the code that creates it.Live scripts are program file that contain your code, output, and formatted text togetherin a single interactive environment called the Live Editor. In live scripts, you can write your code and view the generated output and graphics along with the code that producedit. Add formatted text, images, hyperlinks, and equations to create an interactivenarrative that you can share with others.To diagnose problems in your live scripts or functions, debug your code in the Live Editor. A simple way to determine where a problem occurs in your live script or function is toshow output. To show the output for a line, remove the semi-colon from the end of thatline. The Live Editor displays each output with the line of code that creates it, making iteasy to determine where a problem occurs.

Radar Signal Analysis and Processing Using MATLAB The Elements of MATLAB Style

Annotation This text provides complete, clear, and detailed explanations of the principal numerical analysis methods and well known functions used in science and engineering. These are illustrated with many practical examples. With this text the reader learns numerical analysis with many real-world applications, MATLAB, and spreadsheets simultaneously. This text includes the following chapters:? Introduction to MATLAB? Root Approximations? Sinusoids and Complex Numbers? Matrices and Determinants? Review of Differential Equations? Fourier, Taylor, and Maclaurin Series? Finite Differences and Interpolation? Linear and Parabolic Regression? Solution of Differential Equations by Numerical Methods? Integration by Numerical Methods? Difference Equations? Partial Fraction Expansion? The Gamma and Beta Functions? Orthogonal Functions and Matrix Factorizations? Bessel, Legendre, and Chebyshev Polynomials? Optimization MethodsEach chapter contains numerous practical applications supplemented with detailed instructionsfor using MATLAB and/or Microsoft Excel? to obtain quick solutions.

The Elements of MATLAB Style

Independently Published

Generating code from MATLAB algorithms for desktop and embedded systems allows you to perform your software design, implementation, and testing completely within the MATLAB workspace. You can: Verify that your algorithms are suitable for code generation; Generate efficient, readable, and compact C/C++ code automatically, which eliminates the need to manually translate your MATLAB algorithms and minimizes the risk of introducing errors in the code; Modify your design in MATLAB code to take into account the specific requirements of desktop and embedded applications, such as data type management, memory use, and speed; Test the generated code and easily verify that your modified algorithms are functionally equivalent to your original MATLAB algorithms; Generate MEX functions to cccelerate MATLAB algorithms in certain applications and speed up fixed-point MATLAB code, and Generate hardware description language (HDL) from MATLAB code. To convert MATLAB code to efficient C/C++ code, the code generator introduces optimizations that intentionally cause the generated code to behave differently, and sometimes produce different results, than the original source code. In the MATLAB

language, variables can change their properties dynamically at run time so you can use the same variable to hold a value of any class, size, or complexity. However, statically-typed languages like C must be able to determine variable properties at compile time. Therefore, for C/C++ code generation, you must explicitly define the class, size, and complexity of variables in MATLAB source code before using them. For C/C++ code generation, you should explicitly and unambiguously define the class, size, and complexity of variables before using them in operations or returning them as outputs. Define variables by assignment, but note that the assignment copies not only the value, but also the size, class, and complexity represented by that value to the new variable. When generating C/C++ code from MATLAB, you cannot grow a variable by writing into an element beyond its current size. Such indexing operations produce run-time errors. You must define the matrix first before assigning values to its elements. During C/C++ code generation, the code generator checks for statements that attempt to access uninitialized memory. If it detects execution paths where a variable is used but is potentially not defined, it generates a compile-time error. To prevent these errors, define variables by assignment before using them in operations or returning them as function outputs. Note, however, that variable assignments not only copy the properties of the assigned data to the new variable, but also initialize the new variable to the assigned value. This forced initialization sometimes results in redundant copies in C/C++ code. You can reuse (reassign) an input, output, or local variable with different class, size, or complexity if the code generator can unambiguously determine the properties of each occurrence of this variable during C/C++ code generation. If so, MATLAB creates separate uniquely named local variables in the generated code. You can view these renamed variables in the code generation report. You cannot reuse (reassign) variables if it is not possible to determine the class, size, and complexity of an occurrence of a variable unambiguously during code generation. In this case, variables cannot be renamed and a compilation error occurs.

MATLAB Programming for Biomedical Engineers and Scientists Springer Science & Business Media

This easy-to-follow guide provides academics and industrial engineers with a state-of-the-art numerical approach to the most frequent technical and economical optimization methods. In an engaging manner, it provides the reader with not only a systematic and comprehensive study, but also with necessary and directly implementable code written in the versatile and readily available platform Matlab. The book offers optimization methods for univariate and multivariate constrained or unconstrained functions, general optimization methods and multicriteria optimization methods; provides intuitively, step-by-step explained sample Matlab code, that can be easily adjusted to meet individual requirements; and uses a clear, concise presentation style, which will be suited to readers even without a programming background, as well as to students preparing for examinations in optimization methods.

Signal Processing in MATLAB. Simulink Blocks and Code Generation CRC Press

Generating code from MATLAB

algorithms for desktop and embedded systems allows you to perform your software design, implementation, and testing completely within the MATLAB workspace. You can: -Verify that your algorithms are suitable for code generation-Generate efficient readable, and compact C/C++ code automatically, which eliminates the need to manually translate your MATLAB algorithms and minimizes the risk of introducing errors in the code.-Modify your design in MATLAB code to take into account the specific requirements of desktop and embedded applications, such as data type management, memory use, and speed.-Test the generated code and easily verify that your modified algorithms are functionally equivalent to your original MATLAB algorithms.-Generate MEX functions to: -Accelerate MATLAB algorithms in certain applications.-Speed up fixed-poin MATLAB code.-Generate hardware description language (HDL) from MATLAB code.. To generate C/C++ or MEX code from MATLAB algorithms, you must install the followingsoftware: -MATLAB Coder product-C/C++ compilerWhen writing MATLAB code that you want to convert into efficient standalone C/C++code, you must consider the following: -Data types C and C++ use static typing. To determine the types of your variables before use, MATLAB Coder requires a complete assignment to each variable.-Array sizing Variable-size arrays and matrices are supported for code generation. You can define inputs, outputs, and local variables in MATLAB functions to represent data that varies in size at run time.-MemoryYou can choose whether the generated code uses static or dynamic memory allocation.With dynamic memory allocation, you potentially use less memory at the expense of time to manage the memory. With static memory, you get better speed, but with higher memory usage. Most MATLAB code takes advantage of the dynamic sizing features in MATLAB, therefore dynamic memory allocation typically enables you to generate code from existing MATLAB code without modifying it much. Dynamic memory allocation also allows some programs to compile even when upper bounds cannot be found.Static allocation reduces the memory footprint of the generated code, and therefore issuitable for applications where there is a limited amount of available memory, such asembedded applications.

Matlab Code Generation Springer Nature
Dive into intelligent systems, machine learning, and control with this hands-on, project-based textbook, including over 20 hands-on Arduino, Matlab and Simulink assignments. With over 120 end-of-chapter problems, and solutions for instructors, this is the ideal practical introduction for senior and graduate engineering students.

Practical Optimization with MATLAB Cambridge Scholars Publishing
Imaging modalities in radiology produce ever-increasing amounts of data which need to be displayed, optimized, analyzed and archived: a "big data" as well as an "image processing" problem. Computer programming skills are rarely emphasized during the education and training of medical physicists, meaning that many individuals enter the workplace without the ability to efficiently solve many real-world clinical problems. This book provides a foundation for the teaching and learning of programming for medical physicists and other professions in the field of Radiology and offers valuable content for

novices and more experienced readers alike. It focuses on providing readers with practical skills on how to implement MATLAB® as an everyday tool, rather than on solving academic and abstract physics problems. Further, it recognizes that MATLAB is only one tool in a medical physicist’s toolkit and shows how it can be used as the "glue" to integrate other software and processes together. Yet, with great power comes great responsibility. The pitfalls to deploying your own software in a clinical environment are also clearly explained. This book is an ideal companion for all medical physicists and medical professionals looking to learn how to utilize MATLAB in their work. Features Encompasses a wide range of medical physics applications in diagnostic and interventional radiology Advances the skill of the reader by taking them through real-world practical examples and solutions with access to an online resource of example code The diverse examples of varying difficulty make the book suitable for readers from a variety of backgrounds and with different levels of programming experience.

MATLAB Software for the Code Excited Linear Prediction Algorithm Cambridge Scholars Publishing

Project Report from the year 2018 in the subject Computer Science - Programming, , language: English, abstract: The F5 algorithm proposed by Westfeld is still one of the most known algorithms in the field of DCT-based steganography. It can make a JPEG image a container of a secret message, where no one knows the presence of the message except the sender and the intended receiver. In this programming work, we show how to realize the F5 algorithm via Matlab. We present the block diagrams of embedding and extracting processes and the entire Matlab code of the F5 algorithm. Some Notes about the F5 Matlab code: 1- The implementation code works according to the method proposed by Andreas Westfield in his paper: " F5—A Steganographic Algorithm : High Capacity Despite Better Steganalysis ". Huffman coding and decoding are implemented using the Matlab JPEG Toolbox developed by Phil Sallee. 2- The two-part Matlab code included in the report, embedding and extracting parts, can be executed in Matlab IDE. The embedding part reads the cover JPEG file and the message file we want to hide, then it creates a Stego JPEG file according to the F5 algorithm. On the other side, The extracting part reads the Stego JPEG file, and then it extracts the hidden message file. 3- The F5 code calls the main two functions of Phil Sallee's Matlab Toolbox; JPEG reading and writing. These functions make it easier to access and manipulate the quantized DCT coefficients of a given JPEG file. Using Sallee's Toolbox should accord with the used operating system, whether it is 32 or 64 bits. 4- The F5 code contains the function to form the image matrix to show the input and output images. Running this function requires ALL the Sallee's Toolbox to be installed. Otherwise, the user can REMOVE this function from the code since it doesn't affect the main F5 process and thus keep ONLY using the main two function of the Sallee's Toolbox. 5- The message file we want to hide can be any file of any kind and whatever its extension. The size of the message file should be appropriate for the size of the used cover JPEG image, so no errors will occur when executed.

Matlab für Dummies Scientific e-Resources

Numerical Methods with MATLAB provides a highly-practical reference work to assist anyone working with numerical methods. A wide range of techniques are introduced, their merits discussed and fully working MATLAB code samples supplied to demonstrate how they can be coded and applied. Numerical methods have wide applicability across many scientific, mathematical, and engineering disciplines and are most often employed in situations where working out an exact answer to the problem by another method is impractical. Numerical Methods with MATLAB presents each topic in a concise and readable format to help you learn fast and effectively. It is not intended to be a reference work to the conceptual theory that underpins the numerical methods themselves. A wide range of reference works are readily available to supply this information. If, however, you want assistance in applying numerical methods then this is the book for you.

Numerical Methods using MATLAB CRC Press

An introduction to radar systems should ideally be self-contained and hands-on, a combination lacking in most radar texts. The first edition of Radar Systems Analysis and Design Using MATLAB® provided such an approach, and the second edition continues in the same vein. This edition has been updated, expanded, and reorganized to include advances in the field and to be more logical in sequence. Ideal for anyone encountering the topic for the first time or for professionals in need of on-the-job reference, this book features an abundance of MATLAB programs and code. Radar Systems Analysis and Design Using MATLAB®, Second Edition presents the fundamentals and principles of radar along with enough rigorous mathematical derivations to ensure that you gain a deep understanding. The author has extensively revised chapters on radar cross-section and polarization, matched filter and radar ambiguity function, and radar wave propagation. He also added information on topics such as PRN codes, multipath and refraction, clutter and MTI processing, and high range resolution. With all MATLAB functions updated to reflect version 7.0 and an expanded set of self-test problems, you will find this up-to-date text to be the most complete treatment of radar available, providing the hands-on tools that will enrich your learning.