UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

CURSO DE ENGENHARIA ELETRÔNICA

GUILHERME ALESSANDRO KNOPF

MARIANO EDUARDO BURICH

**ANÁLISE PROBABILÍSTICA DE TEMPOS DE RESPOSTA DE UM BARRAMENTO CAN**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2015

GUILHERME ALESSANDRO KNOPF

MARIANO EDUARDO BURICH

**ANÁLISE PROBABILÍSTICA DE TEMPOS DE RESPOSTA DE UM BARRAMENTO CAN**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia Eletrônica do Departamento Acadêmico de Eletrônica – DAELN – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro Eletrônico.

Orientador: Prof. Msc. Luiz Fernando Copetti

CURITIBA

2015

GUILHERME ALESSANDRO KNOPF

MARIANO EDUARDO BURICH

**ANÁLISE PROBABILÍSTICA DE TEMPOS DE RESPOSTA DE UM BARRAMENTO CAN**

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para obtenção do título de Engenheiro Eletrônico pela Universidade Tecnológica Federal do Paraná.

Curitiba, 26 de agosto de 2015.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Hilton

Coordenador de Curso

Departamento Acadêmico de Eletrônica

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Hilton

Responsável pelo Trabalho de Conclusão de Curso

Departamento Acadêmico de Eletrônica

**BANCA EXAMINADORA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Richard

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Luiz Fernando Copetti

Orientador

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Rebelatto

AGRADECIMENTOS

RESUMO

BURICH, Mariano. KNOPF, Guilherme. **Análise Probabilística de Tempos de Resposta de um Barramento CAN.** 2015. 132 páginas. Trabalho de Conclusão de Curso de Bacharelado em Engenharia Eletrônica – Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

No âmbito de máquinas agrícolas, unidades de controle eletrônico tem sido cada vez mais usadas, para o controle de sistemas hidráulicos complexos, aumentando o tráfego de dados nas linhas de comunicação da rede CAN, o que torna mais difícil a identificação dos tempos de resposta das mensagens. Neste trabalho é desenvolvido um novo método, proposto pela equipe, para identificação das características da rede e dos tempos de resposta das mensagens, utilizando ferramentas de probabilidade e estatística, aplicadas a logs de dados preexistentes de máquinas. O método desenvolvido recebe os logs preexistentes e fornece como resultado curvas de cdf e pdf para cada mensagem, restando ao usuário relacionar a severidade de quebra de deadline com a sua probabilidade de ocorrência, tomando medidas corretivas, se necessário. São também apresentadas ferramentas, presentes em literatura, para identificação de piores tempos de resposta das mensagens e porque elas não são úteis nesse cenário. O trabalho foi aplicado ao log de dados de uma colhedora de cana de açúcar, resultando em uma análise que cumpriu com os objetivos propostos.

**Palavras-chave:** Protocolo CAN, Análise Probabilística, Tempo de Resposta, Colhedora de cana de açúcar.

ABSTRACT

BURICH, Mariano. KNOPF, Guilherme. **Probabilistic Analysis of CAN Bus Response Time.** 2015. 132 pages. Bachelor of Electronics Engineering Final Course Assignment – Federal University of Technology – Paraná. Curitiba, 2015.

In the context of agricultural machinery, electronic control units have been increasingly used for the control of complex hydraulic systems, increasing the data traffic on the communication lines of the CAN network, which makes it difficult to identify the response times of messages. This work developed a new method, proposed by the team, to identify the characteristics of the network and the response times of messages, using tools of probability and statistics, applied to existing machines data logs. The developed method receives the existing logs and provides curves as a result of cdf and pdf for each message, leaving the user to relate the deadline break with their probability of occurrence, taking corrective action if necessary. Tools from literature that identify the worst response times of messages are presented, and it is explained why they are not useful in the studied scenario. The work was applied to the data log of a sugar cane harvester, resulting in an analysis that met the proposed objectives.

**Keywords:** CAN Protocol, Probabilistic Analysis, Response Time, Sugarcane Harvester

LISTA DE ILUSTRAÇÕES

[Figura 1 – Redução de cabeamento. 18](#_Toc436748298)

[Figura 2 – Modelo OSI para CAN. 20](#_Toc436748299)

[Figura 3 - Barramento CAN. 22](#_Toc436748300)

[Figura 4 – Padrão de tensões para o barramento CAN. 23](#_Toc436748301)

[Figura 5 – Analogia ao barramento CAN. 24](#_Toc436748302)

[Figura 6 – Divisão do tempo de bit. 24](#_Toc436748303)

[Figura 7 – Mensagem com *bit-stuffing.* 25](#_Toc436748304)

[Figura 8 – Arbitragem CAN. 26](#_Toc436748305)

[Figura 9 – pacote de dados CAN. 27](#_Toc436748306)

[Figura 10 – Campo de arbitragem padrão e estendido. 28](#_Toc436748307)

[Figura 11 – Campo de controle. 28](#_Toc436748308)

[Figura 12 – campo de dados. 29](#_Toc436748309)

[Figura 13 – A mensagem remota. 31](#_Toc436748310)

[Figura 14 – condição de erro. 32](#_Toc436748311)

[Figura 15 – Identificador J1939. 34](#_Toc436748312)

[Figura 16 – ECUs e barramentos em um veículo. 39](#_Toc436748313)

[Figura 17 – Barramento típico NMEA 40](#_Toc436748314)

[Figura 18 – Display conectado ao barramento CAN. 41](#_Toc436748315)

[Figura 19 – Controles ligados ao barramento CAN. 41](#_Toc436748316)

[Figura 20 – Colhedora de cana de açúcar. 42](#_Toc436748317)

[Figura 21 - Exemplo de arbitragem com 3 mensagens 43](#_Toc436748318)

[Figura 22 - Tempos de Transmissão 44](#_Toc436748319)

[Figura 23 - Transmissão com indicação de início e fim 45](#_Toc436748320)

[Figura 24 - Transmissão com indicação do tempo de invocação da mensagem 2 46](#_Toc436748321)

[Figura 25 - Tempos de invocação coincidiram com o de inicio de transmissão 46](#_Toc436748322)

[Figura 26 - Duas instâncias da mensagem 1 47](#_Toc436748323)

[Figura 27 - Mensagem 2 com invocação e tempo de espera diferente de zero 48](#_Toc436748324)

[Figura 28 - Transmissão com indicação de período entre instâncias 51](#_Toc436748325)

[Figura 29 - Pior tempo de resposta da mensagem 3 em um sistema preemptivo 55](#_Toc436748326)

[Figura 30 - Demonstração da interferência causada na mensagem 3 57](#_Toc436748327)

[Figura 31 - Período ocupado nível 3 59](#_Toc436748328)

[Figura 32 - Período ocupado dividido em janelas de acordo com a tarefa 3 60](#_Toc436748329)

[Figura 33 - Exemplo de escalonamento sem efeito do jitter 63](#_Toc436748330)

[Figura 34 - Exemplo do escalonamento da Figura 33 com efeito do 63](#_Toc436748331)

[Figura 35 - execuções do período 1 não invadem as do periodo 2 65](#_Toc436748332)

[Figura 36 - Início do período ocupado em t=0 por tarefas que sofreram jitter 65](#_Toc436748333)

[Figura 37 - Indicação do pior período ocupado de nível 3 com jitter na tarefa 2 66](#_Toc436748334)

[Figura 38 - Tarefa esporadicamente periódica 67](#_Toc436748335)

[Figura 39 - Pior período ocupado do conjunto dado pela Tabela 8 68](#_Toc436748336)

[Figura 40 - Divisão em janelas da Figura 39 para análise da tarefa 3 68](#_Toc436748337)

[Figura 41 - tempo de invocação de uma mensagem esporadicamente periódica 69](#_Toc436748338)

[Figura 42 - Diferença entre um momento crítico preemptivo e não preemptivo 78](#_Toc436748339)

[Figura 43 - Soma das interferências até wi (I1+I2) resultam em wi 79](#_Toc436748340)

[Figura 44 - Mensagens de maior prioridade (1,2 e 3) sem o tempo de bloqueio 80](#_Toc436748341)

[Figura 45 - Mensagens de maior prioridade (1,2 e 3) bloqueadas 81](#_Toc436748342)

[Figura 46 – Período ocupado de nível 3 gerado pelo momento crítico 3 83](#_Toc436748343)

[Figura 47 - Gráfico representando a relação de probabilidade 87](#_Toc436748344)

[Figura 48 – Tempo ocupado mensagem 4. 89](#_Toc436748345)

[Figura 49 – período da mensagem 2. 90](#_Toc436748346)

[Figura 50 – Tempos de resposta. 91](#_Toc436748347)

[Figura 51 – Conjunto de eventos medidos. 92](#_Toc436748348)

[Figura 52- CDF e PDF para os dados da Tabela 10 97](#_Toc436748349)

[Figura 53 – Período mensagem 1. 98](#_Toc436748350)

[Figura 54 – Invocações com jitter. 99](#_Toc436748351)

[Figura 55 - Modelo de erro de período e de fila 101](#_Toc436748352)

[Figura 56 - Erro de período e erro de fila como gaussianas dobradas 107](#_Toc436748353)

[Figura 57 - Erro gerado pelas imagens da Figura 56 através da equação (120) 108](#_Toc436748354)

[Figura 58 - Pdf ideal do tempo de invocação 111](#_Toc436748355)

[Figura 59 - Projeção dos tempos de invocação 113](#_Toc436748356)

[Figura 60 - Tempo de invocação restritos pelo tempo ocupado 113](#_Toc436748357)

[Figura 61 - Interface gráfica do software MATLAB 118](#_Toc436748358)

[Figura 62 - CANcaseXL - Data logger 119](#_Toc436748359)

[Figura 63 - PDF do tempo de resposta de uma mensagem aperiódica 121](#_Toc436748360)

[Figura 64 - CDF do tempo de resposta de uma mensagem aperiódica 122](#_Toc436748361)

[Figura 65 - PDF do tempo de resposta de uma mensagem periódica 123](#_Toc436748362)

[Figura 66 - CDF do tempo de resposta de uma mensagem periódica 124](#_Toc436748363)

[Figura 67 - Diferença de tempos de resposta 125](#_Toc436748364)

LISTA DE TABELAS

[Tabela 1 - Comprimento sugerido vs. Taxa de transmissão. 22](#_Toc432152261)

[Tabela 2 - Código de comprimento de dados para número de dados 29](#_Toc432152262)

[Tabela 3 - Tempos iniciais e finais de transmissão para cada mensagem *k*  e instância *j* 47](#_Toc432152263)

[Tabela 4 - Tempos estimados de invocação com base na terceira instância 51](#_Toc432152264)

[Tabela 5 - Exemplo de conjunto de tarefas periódicas em um SO 54](#_Toc432152265)

[Tabela 6 - Conjunto de tarefas periódicas em um SO onde a tarefa 3 tem um tempo de resposta maior que o seu período 59](#_Toc432152266)

[Tabela 7 – Detalhe dos tempos fornecidos na Figura 31 59](#_Toc432152267)

[Tabela 8 - Conjunto de tarefas onde duas são esporadicamente periódicas 67](#_Toc432152268)

[Tabela 9 - Conjunto de mensagens que tem falha na análise do tempo de resposta 83](#_Toc432152269)

[Tabela 10- Exemplo de transmissão com tempo ocupado 96](#_Toc432152270)

[Tabela 11 - Relatório de saída de uma análise de barramento através do método proposto 129](#_Toc432152271)

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ABS – Anti-brake system

ACK – Acknowledge

ADC – Analog Digital Converter

CAN – Controller Area Network

CDF – Cumulative Density Function

CiA – CAN in Automation

CNH – Case New Holland

CRC – Cyclic Redundancy Check

CSMA/AMP – Carrier sense multiple access with Arbitration on Message Priority

DLC – Data Length Code

ECU – Electronic Controller Unit

EOF – End of Frame

FMS – Fleet Management System

GPS – Global positioning system

ISI – Intersymbol Interference

ISO – International Organization for Standardization

LLC – Logical Link Control

MAC – Media Access Control

NMEA – National Marine Electronics Association

NRZ – Non-Return to Zero

OSI – Open Systems Interconnection

PDF – Probability Density Function

PDU – Parameter Data Unit

PG – Page Group

PGN – Parameter Group Number

RTR – Remote Transmission Request

SAE – Society of Automotive Engineers

SO – Sistema Operacional

SOF – Start of Frame

SRR – Substitute remote request

SUMÁRIO

[1 INTRODUÇÃO 13](#_Toc435992132)

[1.1 MOTIVAÇÃO 14](#_Toc435992133)

[1.2 ESTRUTURA DO DOCUMENTO 16](#_Toc435992134)

[2 PROTOCOLO CAN 18](#_Toc435992135)

[2.1 HISTÓRICO 18](#_Toc435992136)

[2.2 ASPECTOS GERAIS 19](#_Toc435992137)

[2.3 ARQUITETURA 20](#_Toc435992138)

[2.4 CAMADA FÍSICA 21](#_Toc435992139)

[2.4.1 Meio de transmissão 21](#_Toc435992140)

[2.4.2 Nível de sinais 23](#_Toc435992141)

[2.4.3 Temporização de bit 24](#_Toc435992142)

[2.4.4 Bit-Stuffing 25](#_Toc435992143)

[2.5 CAMADA DE ENLACE 25](#_Toc435992144)

[2.5.1 Arbitragem 26](#_Toc435992145)

[2.5.2 Mensagem CAN 27](#_Toc435992146)

[2.5.2.1 SOF 27](#_Toc435992147)

[2.5.2.2 Campo de arbitragem 27](#_Toc435992148)

[2.5.2.3 Campo de controle 28](#_Toc435992149)

[2.5.2.4 Dados 29](#_Toc435992150)

[2.5.2.5 CRC 29](#_Toc435992151)

[2.5.2.6 ACK 30](#_Toc435992152)

[2.5.2.7 EOF 30](#_Toc435992153)

[2.5.3 Tipos de mensagens 31](#_Toc435992154)

[2.5.3.1 Mensagem de dados 31](#_Toc435992155)

[2.5.3.2 Mensagem remota 31](#_Toc435992156)

[2.5.3.3 Mensagem de erro 32](#_Toc435992157)

[2.5.3.4 Mensagem de Sobrecarga 33](#_Toc435992158)

[3 PADRÃO J1939 34](#_Toc435992159)

[3.1 FORMATO DE MENSAGENS 34](#_Toc435992160)

[3.2 PROTOCOLO DE TRANSPORTE 35](#_Toc435992161)

[3.3 ENDEREÇAMENTO 36](#_Toc435992162)

[4 APLICAÇÕES DO PROTOCOLO CAN 38](#_Toc435992163)

[4.1 AUTOMAÇÃO 38](#_Toc435992164)

[4.2 AUTOMOTIVA 38](#_Toc435992165)

[4.3 NÁUTICA 39](#_Toc435992166)

[4.4 AGRÍCOLA 40](#_Toc435992167)

[5 ATRASO DO TRÁFEGO DE MENSAGENS NO BARRAMENTO CAN 43](#_Toc435992168)

[5.1 TEMPO DE TRANSMISSÃO 44](#_Toc435992169)

[5.2 TEMPOS DE INÍCIO E FIM DE TRANSMISSÃO 44](#_Toc435992170)

[5.3 TEMPO DE INVOCAÇÃO 45](#_Toc435992171)

[5.4 INSTÂNCIA DE UMA MENSAGEM 46](#_Toc435992172)

[5.5 TEMPO DE RESPOSTA 48](#_Toc435992173)

[5.6 MENSAGENS PERIÓDICAS 49](#_Toc435992174)

[5.6.1 Distância entre instâncias ou invocações 50](#_Toc435992175)

[5.6.2 Período 50](#_Toc435992176)

[6 ANÁLISE DO TEMPO DE RESPOSTA 52](#_Toc435992177)

[6.1 ESCALONAMENTO DE SISTEMAS OPERACIONAIS 53](#_Toc435992178)

[6.1.1 Relação com sistemas operacionais de prioridade fixa 53](#_Toc435992179)

[6.1.2 Exemplo de aplicação da equação de Joseph e Pandya 57](#_Toc435992180)

[6.1.3 Conceito de busy-period 58](#_Toc435992181)

[6.1.4 Equação genérica para deadline arbitrário 60](#_Toc435992182)

[6.1.5 Tempo de bloqueio 62](#_Toc435992183)

[6.1.6 Jitter de soltura 62](#_Toc435992184)

[6.1.7 Tarefas esporadicamente periódicas 67](#_Toc435992185)

[6.1.7.1 Tempos de invocação 68](#_Toc435992186)

[6.1.7.2 Interferência de mensagens mais prioritárias 70](#_Toc435992187)

[6.1.7.3 Interferência da mesma mensagem 71](#_Toc435992188)

[6.1.7.4 Fim da janela 72](#_Toc435992189)

[6.1.7.5 Tempo de resposta 72](#_Toc435992190)

[6.1.7.6 Jitter 73](#_Toc435992191)

[6.1.8 Teste completo de escalonabilidade 74](#_Toc435992192)

[6.2 ADAPTAÇÃO DAS EQUAÇÕES DE ESCALONAMENTO AO BARRAMENTO CAN 76](#_Toc435992193)

[6.2.1 Diferenças entre preemptivo e não preemptivo 76](#_Toc435992194)

[6.2.2 Tempo de transmissão 78](#_Toc435992195)

[6.2.3 Interferência de mensagens mais prioritárias 79](#_Toc435992196)

[6.2.4 Tempo de bloqueio 80](#_Toc435992197)

[6.2.5 Jitter e tempo de bit 81](#_Toc435992198)

[6.2.6 Falha na análise de tempo de resposta 82](#_Toc435992199)

[6.2.7 Correção na análise de tempo de resposta 84](#_Toc435992200)

[6.2.7.1 Período Ocupado 84](#_Toc435992201)

[6.2.7.2 Fim do período ocupado 85](#_Toc435992202)

[6.2.7.3 Tempo de resposta de cada janela 85](#_Toc435992203)

[6.3 CARACTERÍSTICAS DA ANÁLISE DE PIOR TEMPO DE RESPOSTA 86](#_Toc435992204)

[6.4 ANÁLISE ESTATÍSTICA DO BARRAMENTO CAN 88](#_Toc435992205)

[6.4.1 Tempo ocupado 89](#_Toc435992206)

[6.4.2 Método 90](#_Toc435992207)

[6.4.2.1 Relação entre o tempo ocupado e a probabilidade do tempo de resposta 91](#_Toc435992208)

[6.4.2.2 Modelo de PDF para mensagens aperiódicas 95](#_Toc435992209)

[6.4.2.3 Modelo de PDF para mensagens sem tempo ocupado 95](#_Toc435992210)

[6.4.2.4 Exemplo de aplicação em uma mensagem não periódica 96](#_Toc435992211)

[6.4.3 Estimativa do período 97](#_Toc435992212)

[6.4.4 Jitter 99](#_Toc435992213)

[6.4.4.1 Modelo de erro 100](#_Toc435992214)

[6.4.4.2 Período 102](#_Toc435992215)

[6.4.4.3 PDF do Jitter através de invocações sucessivas 103](#_Toc435992216)

[6.4.4.4 PDF do Jitter através de invocações distantes 105](#_Toc435992217)

[6.4.4.5 Variância do erro de período 106](#_Toc435992218)

[6.4.4.6 Projeção da pdf de tempo de invocação 110](#_Toc435992219)

[6.4.4.7 Tempo ocupado na estimativa da invocação de uma mensagem periódica 112](#_Toc435992220)

[6.4.5 Mensagens esporadicamente periódicas 114](#_Toc435992221)

[6.4.6 Sumário do método da estimativa de pdf 116](#_Toc435992222)

[7 IMPLEMENTAÇÃO E RESULTADOS 118](#_Toc435992223)

[7.1 LOGS UTILIZADOS 119](#_Toc435992224)

[7.2 RESULTADOS 120](#_Toc435992225)

[7.2.1 Mensagens aperiódicas 120](#_Toc435992226)

[7.2.2 Mensagens periódicas 122](#_Toc435992227)

[7.2.3 Influência da prioridade de uma mensagem 124](#_Toc435992228)

[8 CONSIDERAÇÕES FINAIS 126](#_Toc435992229)

[8.1 COMPARAÇÃO ENTRE MÉTODOS 126](#_Toc435992230)

[8.2 CARACTERÍSTICAS DO SOFTWARE DESENVOLVIDO 126](#_Toc435992231)

[8.3 INTERPRETAÇÃO DO RELATÓRIO FINAL 127](#_Toc435992232)

[8.4 TRABALHOS FUTUROS 127](#_Toc435992233)

[9 Referências 132](#_Toc435992234)

# INTRODUÇÃO

Com a popularização da eletrônica embarcada, sistemas computadorizados estão sendo cada vez mais incorporados a diversos tipos de veículos (ônibus, caminhões, carros, colheitadeiras de grãos) tendo diferentes objetivos em função das finalidades e necessidades de cada veículo, muitas vezes constituindo em um sistema invisível ao operador.

Um dos objetivos da implantação desses sistemas eletrônicos é otimizar e possibilitar processos técnicos realizados pelo próprio veículo, como realizar o controle retroalimentado de uma ferramenta de corte, injetar a quantidade de combustível exata no motor e percorrer uma rota previamente definida em função de coordenadas GPS, permitindo a sua utilização como ferramentas altamente competitivas e eficazes, fazendo-as ganhar um espaço muito significativo em certos setores econômicos, como por exemplo, o agrícola.

Os sistemas embarcados veiculares também são utilizados com o intuito de fornecer segurança e conforto aos tripulantes de um veículo, através de sistemas como o freio com ABS, avisos de porta-aberta, airbag e piloto automático; gerando a inclusão de complexos sistemas eletrônicos em máquinas que antes eram quase inteiramente mecânicas.

A fim de introduzir modularidade, diminuir os custos e integrar mais facilmente diversos fabricantes, os sistemas computadorizados de um veículo não funcionam em um único microprocessador, mas em diversos módulos (ECU), cada qual com o seu processador, realizando o seu conjunto de tarefas específico para a sua funcionalidade.

Porém, a modularidade introduz a necessidade de um mecanismo de comunicação, uma vez que cada módulo pode depender de informações ou eventos que são conhecidos somente por outro. Como por exemplo, veículos agrícolas que através do módulo do display informam erros e alarmes que ocorreram em sistemas internos do motor, auxiliando o operador a agir de acordo com a falha e fornecendo aos mantenedores indícios exatos do problema.

Devido à dificuldade práticas, como o estabelecimento de conexões ponto-a-ponto entre todas as ECUs, foi percebida a necessidade da criação de um protocolo de comunicação que fosse robusto (ambiente ruidoso), flexível (módulos podem ser conectados e desconectados em tempo real) e veloz, levando à criação do protocolo CAN pela Bosch.

O barramento CAN tem a característica de permitir que somente um nó transmita por vez, através de um processo de arbitragem. Este fornece o direito de uso do barramento ao módulo que deseja enviar a mensagem que tem a maior prioridade no momento, introduzindo um tempo de espera em mensagens dos módulos que não puderam utilizar o barramento.

Devido ao fato desse sistema de comunicação operar em tempo real, semelhante ao escalonamento de um sistema operacional, cada mensagem tem um tempo limite (*deadline*) para ser enviado sem que ocorram efeitos indesejados. Este limite não pode ser ultrapassado pelo tempo de resposta da própria mensagem. Os efeitos indesejados de um estouro de *deadline* podem ser diversos e dependem de qual ECU e de qual mensagem ultrapassou o limite de tempo. Isto pode ocasionar, por exemplo, uma variação na taxa de atualização da velocidade do veículo mostrado em um display, uma piora na performance do veículo através de um comando do joystick que foi atrasado pelo barramento, ou até mesmo uma colisão de alguma parte da máquina pois um sensor de proximidade que não teve o seu sinal enviado no tempo correto.

Esses efeitos indesejados servem como motivação para este trabalho. Ele consiste na proposição de um método de análise dos tempos de resposta do barramento CAN, fornecendo curvas de probabilidade de estouro de *deadline*. Estas podem ser utilizadas para determinar a confiabilidade e eficiência da rede de comunicação implantada em um veículo, gerando mais segurança sobre o produto final e por fim, tornando-o mais competitivo.

Um cenário de teste é extremamente interessante para este tipo de análise. Desta forma, a empresa CNH Industrial, buscando aumentar a confiabilidade de seus produtos e obter conhecimento técnico neste ramo, forneceu todas as informações necessárias sobre a rede CAN da máquina colhedora de cana de açúcar da marca Case IH. Isto auxiliou o desenvolvimento e aplicação do software de análise de tempo de resposta descrito neste documento.

## MOTIVAÇÃO

O protocolo CAN é amplamente utilizado em diversos veículos, como a colhedora de cana de açúcar da marca Case IH, pois consiste em um método robusto e de baixo custo para comunicação de boa qualidade entre diversas ECUs.

Devido ao seu modo de funcionamento, este sistema de comunicação insere atrasos não determinísticos às mensagens que transitam pelo barramento, o que pode gerar falhas nos sistemas que operam no veículo.

Essas falhas têm diversos graus de severidade. O atraso de uma mensagem que identifica um erro de um sensor pode ser considerada de baixa severidade, assim como um atraso na comunicação com o módulo do gps, que pode demorar em corrigir a rota da máquina, pode ser visto como um erro de média severidade.

Apesar das falhas citadas anteriormente serem importantes, a maior preocupação está na possibilidade de ocorrência de falhas de grande severidade, as quais colocam em risco a integridade física das pessoas.

O exemplo mais evidente de uma falha de grande severidade é o atraso na resposta de um comando no joystick de direcionamento da máquina, pois a mesma precisa ser rápida o suficiente para obedecer uma manobra de emergência realizada pelo operador, evitando uma colisão com objetos e com alguma pessoa que acidentalmente se coloque em uma posição de perigo em frente ao veículo.

Apesar desse tipo de falha ser raro, pois os veículos são rigorosamente testados através de uma série de processos, é interessante realizar uma análise para verificar se existe a possibilidade desse tipo de erro acontecer, e quão segura é a margem de operação do barramento.

Pode existir também o interesse do fabricante do veículo alterar o software de algum módulo incluindo uma nova funcionalidade, acarretando em novas mensagens no barramento. Não existe a necessidade do teste completo do veículo devido à inclusão de uma nova mensagem no barramento, porém, deve-se quantificar o impacto dessa mensagem no barramento.

Esse impacto é atualmente quantificado pela métrica de utilização total do barramento, a qual não deve passar de 40%. Esse valor foi difundido na indústria através dos anos como correto, entretanto, não possui uma origem científica e acaba sendo utilizado por ter funcionado empiricamente. Muitas vezes duas redes CAN são instaladas em um veículo para atender a esse limite.

Ao utilizar o limiar de 40% sem um conhecimento teórico, pode-se estar superdimensionando o barramento e gerando custos excessivos sem necessidade alguma.

A partir desses fatos, a empresa CNHi teve interesse em aprofundar-se no assunto, gerando o estudo descrito neste documento, onde é realizada uma análise bibliográfica e uma proposta de metodologia para analisar os tempos de resposta de um barramento CAN. O objetivo deste estudo é fornecer à empresa uma maneira de melhorar a confiabilidade e segurança dos seus produtos, diminuir os custos e possuir um conhecimento diferenciado em relação aos seus concorrentes.

O objetivo específico é desenvolver uma ferramenta de análise probabilística de tempos de resposta de um barramento CAN.

## ESTRUTURA DO DOCUMENTO

No início de cada seção existe um breve resumo de seus objetivos, afim de auxiliar o leitor que já tenha domínio sobre determinado assunto, como as especificações do protocolo CAN ou a análise teórica do pior tempo de resposta, a localizar mais facilmente as seções de seu interesse.

A seção inicial deste documento tem como objetivo ambientar o leitor aos problemas relacionados ao protocolo CAN e seus impactos, levando ao estabelecimento da motivação e importância da análise dos tempos de resposta.

As seções seguintes (2 e 3) descrevem a especificação dos protocolos utilizados na máquina colhedora de cana de açúcar da CNHi, sendo estes o CAN e o J1939 (baseado em CAN), respectivamente.

A seção 4 descreve aplicações onde o protocolo CAN é utilizado amplamente, e aos quais o estudo realizado pode ser direcionado.

Na quinta seção (seção 5) são definidas e explicadas as relações entre as variáveis matemáticas e os seus significados físicos no barramento, como o tempo de execução e de invocação das mensagens, as quais são abordadas em todos os capítulos subsequentes.

A seção 0 consiste na parte técnica mais importante do documento, onde é realizada uma extensa análise bibliográfica em 6.1 e 6.2, introduzindo conceitos e métodos não probabilísticos para a análise de pior tempo de resposta retirados da bibliografia.

Em 6.3 são demonstrados motivos pelos quais a análise realizada pela bibliografia não supre as necessidades práticas criadas pela motivação do estudo, levando à necessidade de uma outra abordagem do problema.

A seção 6.4 consiste na contribuição original deste documento, onde é desenvolvida uma análise probabilística dos tempos de resposta, através da utilização de *logs* de dados do barramento. O método desenvolvido fornece curvas de *pdf* e *cdf* para cada mensagem pertencente ao barramento, assim como um relatório final de dados obtidos pelo algoritmo.

A análise das curvas fornecidas não é escopo do projeto, pois as possíveis falhas geradas devido à um atraso na comunicação entre ECUs são muito particulares ao modo em que os sistemas do veículo foram arquitetados. A necessidade de relacionar a origem das falhas com o atraso de uma determinada mensagem é proposto como um trabalho futuro.

Na seção 7 são discutidas a implementação e os resultados, enquanto que na seção 8 são realizadas as considerações finais.

# PROTOCOLO CAN

O protocolo CAN (Controller Area Network) é um padrão de barramento serial de múltiplos mestres utilizado em veículos, que possibilita a comunicação entre diferentes unidades eletrônicas distribuídas no mesmo.

Nesta seção serão abordados aspectos históricos, físicos e lógicos da especificação do protocolo CAN, com o objetivo de fornecer ao leitor uma maior familiaridade com a necessidade de utilização e características do protocolo.

## HISTÓRICO

No passado as montadoras de veículos usavam conexões ponto-a-ponto entre os diversos controladores eletrônicos dentro do veículo. Ao passar do tempo, cada vez mais controladores eletrônicos eram adicionados aos veículos, resultando em cabeamentos mais pesados, volumosos e complexos.

A fim de reduzir custo, peso e complexidade deste sistema, foram desenvolvidas redes internas nos veículos para substituir os cabeamentos dedicados que interligavam os controladores. O barramento CAN surgiu a partir destas necessidades do mercado, entregando robustez e baixo custo.

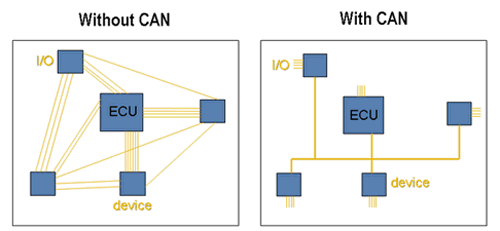


Figura 1 – Redução de cabeamento.

FONTE: http://www.ni.com/white-paper/2732/en/

O protocolo CAN foi desenvolvido inicialmente pela Robert Bosch GmbH em 1983 e apresentado internacionalmente em 1986 no congresso de engenharia automotiva SAE. O barramento foi altamente aceito e em 1993 virou um padrão internacional, ISO 11898.

As primeiras empresas a apoiar o protocolo CAN foram Intel e Philips, que desenvolveram os controladores CAN 82526 e 82C200, respectivamente, e os lançaram em 1987.

O primeiro veículo a usar uma rede interna baseada em CAN foi a BMW 850, lançado em 1988, reduzindo o cabeamento em 2 km, o peso do veículo em 50 kg e usando somente metade dos conectores usados em outros veículos (JOHANSSON, 2000).

Em 1992 foram lançados os primeiros carros da Mercedes-Benz a utilizarem o barramento CAN completo.

Os carros de hoje em dia apresentam dezenas de controladores eletrônicos para diversos fins como controle de velocidade ou monitoramento de RPM do motor, todos interligados através da rede CAN. Através da rede CAN é possível diagnosticar problemas no veículo com maior facilidade.

O protocolo CAN é um dos cinco protocolos usados no padrão de diagnóstico de veículos OBD-II. O padrão OBD-II é obrigatório para todos os carros desde 1996, nos E.U.A. E o padrão EOBD, versão europeia do OBD-II, também é obrigatório a carros desde 2001, na União europeia.

A partir de 1994, vários outros protocolos de alto-nível foram padronizados com base no protocolo CAN, como o CANOpen, DeviceNet, ISOBUS, CAN Kingdom, EnergyBus.

## ASPECTOS GERAIS

Em termos gerais o barramento CAN é uma rede de baixo custo e baixo peso, por apresentar somente um ou dois fios interligando todos os controladores presentes na rede, apresenta comunicação em broadcast, onde todos os controladores presentes na rede podem ouvir o que está sendo transmitido e cabe a cada controlador decidir qual mensagem é destinada a ele. O CAN é um barramento de múltiplos mestres, onde cada controlador, também chamado de nó, pode transmitir no barramento sem interferências, sendo arbitrado somente pelas prioridades das mensagens.

As mensagens do barramento CAN apresentam no início de sua transmissão o campo de arbitragem, que contém a prioridade de cada mensagem para que seja feita a arbitragem do barramento, esta arbitragem é não destrutiva e permite a transmissão não interrompida das mensagens com maior prioridade.

O protocolo CAN também contém o código de redundância cíclica (CRC), que realiza a verificação de erros no conteúdo de cada frame, as mensagens que apresentarem erros são ignoradas pelos nós e um frame de erro é enviado para sinalizar erro no barramento. Se muitos frames de erro forem enviados ao barramento os nós podem parar de enviar frames de erro ou até mesmo podem se desconectar do barramento.

## ARQUITETURA

O protocolo CAN foi especificado abrangendo duas camadas do modelo OSI/ISO, sendo eles a camada física e a de enlace. O modelo OSI é um modelo padrão de referência para os diversos protocolos de comunicação entre sistemas que especifica sete camadas básicas para os protocolos de comunicação, sendo eles: física, enlace, rede, transporte, sessão, apresentação e aplicação.

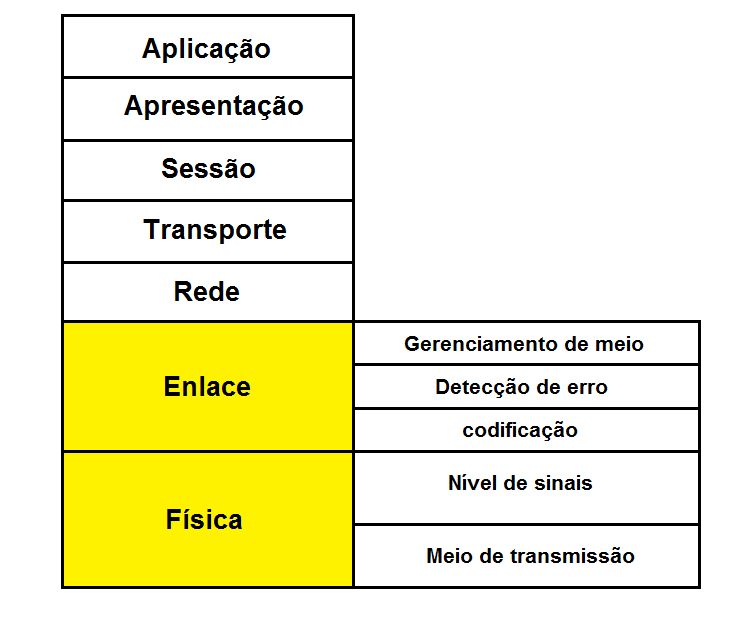


Figura 2 – Modelo OSI para CAN.

Fonte: Autoria própria.

A camada física do modelo define as características físicas da linha de transmissão da rede, incluindo layout de pinos, tensões, impedância da linha, especificação de cabos, temporização de sinais.

A camada de enlace define filtros de aceitação, gerenciamento de acesso ao barramento, detecção de erro, codificação.

## CAMADA FÍSICA

O padrão ISO 11898 define a camada física e a camada de enlace. Para a camada física existem três padrões principais, ISO 11898-1, ISO 11898-3 e J2411.

### Meio de transmissão

Uma das vantagens da linha de transmissão CAN é que ela pode se utilizar de par trançado diferencial ou um fio de cobre. As linhas diferenciais utilizam par trançado de impedância característica de 120Ω. Terminadores resistivos nas pontas da rede com mesmo valor da impedância característica devem ser usados para casar com a impedância da linha e não causar reflexões no barramento.

A rede CAN foi concebida como um barramento, onde existe uma linha de dados principal e todos os nós são conectados diretamente a este barramento. Este tipo de rede tem a vantagem de não precisar de vários fios interligando cada nó do sistema. A figura a seguir exemplifica o barramento CAN.

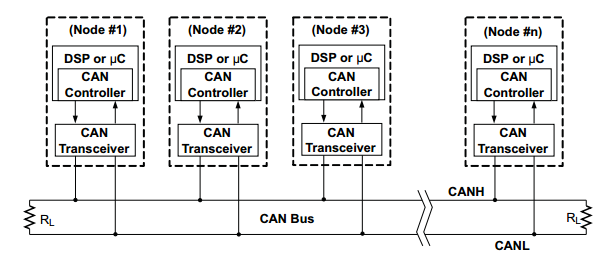


Figura 3 - Barramento CAN.

Fonte: http://www.ti.com/lit/an/slla270/slla270.pdf

As especificações ISO 11898 determinam que a taxa máxima de sinal no barramento é de 1 Mbps, comprimento máximo de 40 m e um máximo de 30 nós. Além disso, determina que o comprimento máximo de braços não terminados deva ser de 0,3 m.

Para que sejam alcançados comprimentos maiores que 40 m, devem ser usadas taxas menores de transmissão. A maior limitação ao comprimento da linha é devido ao atraso de propagação dos sinais dentro do barramento. Outros fatores que influenciam na restrição de comprimento seriam a limitação de banda da linha, causando interferência entre símbolos (ISI), perda de amplitude do sinal devido à resistência dos cabos, efeito pelicular, perda de dielétrico.

A seguinte tabela relaciona os comprimentos de barramento usados para cada taxa de transmissão.

Tabela 1 - Comprimento sugerido vs. Taxa de transmissão.

|  |  |
| --- | --- |
| Comprimento do barramento (m) | Taxa de transmissão (Mbps) |
| 40 | 1 |
| 100 | 0.5 |
| 200 | 0.25 |
| 500 | 0.125 |
| 1000 | 0.05 |

Fonte: http://www.ti.com/lit/an/slla270/slla270.pdf

No barramento CAN a maioria das aplicações utiliza par trançado diferencial para transmissão dos dados, pois esta solução possui baixo custo e boa imunidade a ruído. As duas linhas de sinal do par trançado são denominadas de CANL e CANH.

### Nível de sinais

A norma ISO 11898 especifica dois estados lógicos para os níveis de barramento, as tensões diferenciais da linha devem representar o estado (ou ‘bit’) recessivo e o estado dominante. A figura a seguir ilustra os níveis de tensão de cada estado.

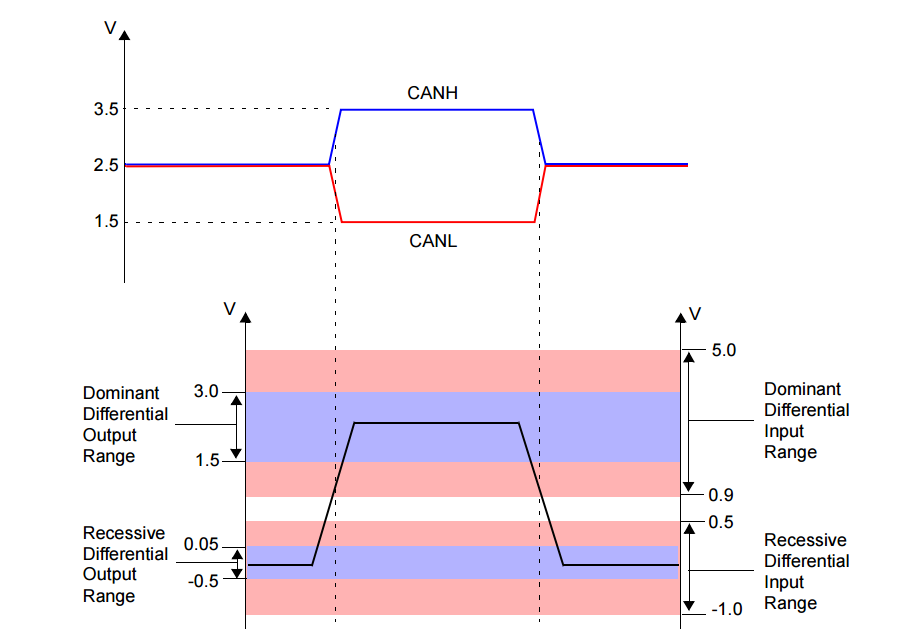


Figura 4 – Padrão de tensões para o barramento CAN.

Fonte: http://ww1.microchip.com/downloads/en/AppNotes/00228a.pdf

O estado recessivo é representado pelo bit lógico ‘1’, a tensão diferencial da linha deve ser menor que um limiar de 0.45V para o transmissor e de 1.5V para o receptor. A tensão das linhas CANH e CANL deve ser aproximadamente igual a 2.5V, para que a tensão diferencial seja 0V.

No estado dominante, representado pelo bit lógico ‘0’, a tensão diferencial deve ser maior que o limiar mínimo. A tensão dos fios CANH e CANL deve ser diferente e o CANH deve ser maior que o CANL. O bit dominante deve sobrescrever não destrutivamente o bit recessivo no barramento para que a arbitragem bit-a-bit seja possível.

O barramento CAN pode ser analogamente comparado a um circuito com transistores como o da figura a seguir.

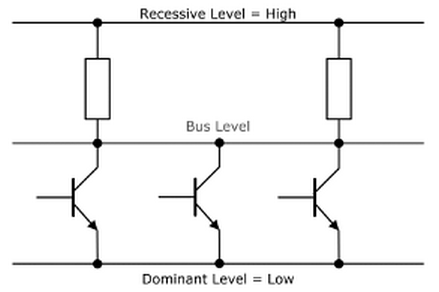


Figura 5 – Analogia ao barramento CAN.

Fonte: A Comprehensible Guide to Controller Area Network Por Wilfried Voss.

A figura mostra três nós, representados por três transistores em modo coletor aberto (*And* por fios). O nível do barramento será dominante (nível baixo) se qualquer um dos transistores tiver um nível baixo. O barramento só apresentara nível alto se todos os transistores estiverem no nível alto.

### Temporização de bit

Cada bit dos sinais do barramento CAN é dividido em quatro partes, o ponto de amostragem do bit é geralmente em 75% do tempo de bit. A figura a seguir mostra as quatro partes do bit.

Sincronismo

Propagação

Fase 1

Fase 2

Tempo de bit

Amostragem

Figura 6 – Divisão do tempo de bit.

Fonte: Autoria própria.

O primeiro segmento, a fase de sincronismo, é o tempo reservado para que uma transição de recessivo para dominante ocorra, na borda de subida, onde os nós devem ser sincronizados. O segundo segmento é a fase de propagação, que reserva um determinado tempo para que os atrasos de transmissão sejam compensados.

Os segmentos das Fases 1 e 2 são usados para a ressincronização, onde a fase 1 deve ser encurtada ou a fase 2 deve ser alargada. A amostragem do bit ocorre imediatamente após a fase 1.

### Bit-Stuffing

Os módulos do barramento CAN utilizam das transições do barramento para realizarem a sincronia de seus *clocks.* Se uma sequência grande de bits de mesma polaridade for enviada é possível que os módulos percam a sincronia, pois os bits são codificados no formato NRZ (*Non-Return to Zero*). Para que isto não ocorra é aplicado o *bit-stuffing.*

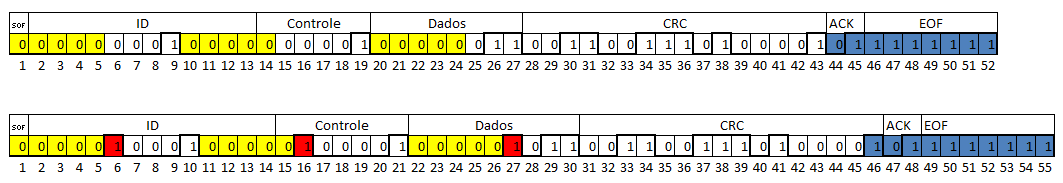


Figura 7 – Mensagem com *bit-stuffing.*

Fonte: Autoria própria.

O *bit-stuffing* é a adição de um bit de polaridade contraria ao final de uma sequência de cinco bits de mesma polaridade, para que os módulos possam sincronizar-se corretamente, os campos de ACK, EOF e delimitador de CRC são ignorados no *bit-stuffing*. Na Figura 7 é mostrado um exemplo de *bit-stuffing* de uma mensagem com sequencias de bits ‘0’, os bits adicionados pelo *bit-stuffing* estão destacados em vermelho, a sequência de zeros em amarelo e os bits não afetados em azul.

## CAMADA DE ENLACE

A camada de enlace pode ser dividida em duas subcamadas, sendo elas LLC (*Logical Link Control* ) e MAC (*Medium Access Control*). A subcamada LLC é responsável por filtrar as mensagens (de acordo com o ID), notificações de sobrecarga de barramento e gerenciamento de recuperação. A camada MAC é responsável pelo encapsulamento de dados, codificação de frames, detecção de erros e serialização de mensagens.

O protocolo CAN opera em broadcast, ou seja, todos os nós da rede podem ouvir os dados sendo transmitidos no barramento e fica a critério de cada nó decidir quais mensagens devem ser filtradas ou não.

### Arbitragem

O acesso simultâneo por diferentes nós ao barramento é permitido pelo protocolo CAN. Quando mais de um nó precisa transmitir informações ao mesmo tempo é necessário um método de resolução de colisões. A resolução de colisões no barramento CAN é feita através de arbitragem, onde a mensagem com o menor ‘ID’ tem maior prioridade e pode transmitir antes.

Cada nó que escreve no barramento o lê de volta, bit a bit, verificando se o que está sendo enviado é igual ao que está sendo lido. No barramento os bits recessivos são sobrescritos por bits dominantes. Se um dos bits enviados no barramento for diferente durante a arbitragem, então o nó que está enviando e percebeu a diferença para de enviar e fica esperando pelo barramento vazio. Este método de arbitragem é chamado de “Carrier Sense Multiple Access with Arbitration on Message Priority” CSMA/AMP (CAN in Automation (CiA), 2002). A imagem abaixo exemplifica o momento de arbitragem.

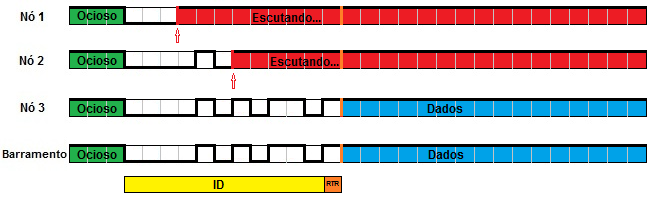


Figura 8 – Arbitragem CAN.

Fonte: Autoria própria.

No exemplo da figura acima, o nó um perde a arbitragem do barramento, pois seu quinto bit é recessivo, ou seja, seu ID é maior que outros, o nó dois também perde o barramento por ter prioridade menor que o nó três. O nó três ganha o barramento e pode enviar os seus dados, pois tem o menor ID dos três, ou a maior prioridade, e continua enviando o conteúdo de sua mensagem. Assim o nó com a maior prioridade sempre manda sua mensagem antes e os outros nós esperam o completo envio da mensagem mais prioritária para enviar suas mensagens.

### Mensagem CAN

Na Figura 9 é apresentada a estrutura do frame de mensagens do protocolo CAN. Uma mensagem inicia com o início de frame (SOF), logo após vem o campo de arbitragem, o campo de controle, os dados, o CRC, o campo de confirmação (ACK) e o fim de frame (EOF).

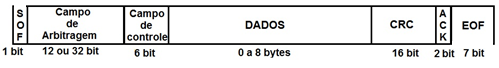


Figura 9 – pacote de dados CAN.

Fonte: Autoria própria.

#### SOF

O início de frame SOF é um indicador de início de transmissão com valor sempre dominante, mas também é usado para fazer o sincronismo entre os nós do barramento. Toda vez que ocorrer uma transição de bit recessivo para dominante enquanto o barramento está ocioso os nós devem sincronizar.

#### Campo de arbitragem

O ID é o primeiro campo de frame de mensagens CAN, que também pode ser chamado de campo de arbitragem. Cada mensagem possui um ID único, pois nesse campo é realizada a arbitragem.

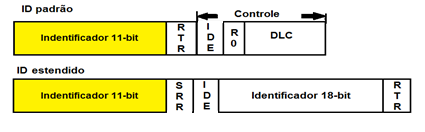


Figura 10 – Campo de arbitragem padrão e estendido.

Fonte: Autoria própria.

O tamanho do ID no formato padrão é de 11 bits, sendo ele o ID base no formato estendido. Após o ID vem o bit RTR que em mensagens de dados deve ser um bit dominante, ao passo que, em pacotes remotos, esse bit deve ser recessivo. O bit RTR representa uma mensagem remota de pedido de retransmissão. Após o ID base aparece o bit IDE, identificador de extensão, deve ser um bit dominante para o formato padrão e recessivo para o formato estendido.

O formato estendido é composto por duas partes, uma com o ID base e a outra com o ID estendido. O bit SRR é o substituto para o RTR no padrão estendido e deve ser um bit recessivo. Os receptores ignoram este valor, mas ele é considerado na arbitragem e no bit-stuffing.

#### Campo de controle

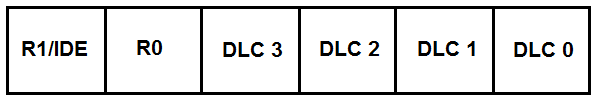


Figura 11 – Campo de controle.

Fonte: Autoria própria.

O campo de controle do frame CAN consiste em seis bits e seu conteúdo é ligeiramente diferente no formato padrão e no formato estendido. O primeiro bit representa o bit de IDE para o formato padrão enquanto no estendido é o bit reservado um. Os bits reservados, representados por ‘R0’ e ‘R1’ na Figura 11, devem ser enviados como bits dominantes. A região dos bits de DLC (Data Length Code) do campo de controle representa a quantidade de bytes de dados a serem enviados no pacote da mensagem.

O código de comprimento de dados, DLC, presente no campo de controle é um código que, como o nome diz, indica a quantidade de bytes de dados a serem transmitidos pela mensagem. A quantidade de bytes pode variar de zero a oito bytes de dados e o código, que contém quatro bits, vai do valor zero a sete. Todos os outros valores do código representam oito bytes de dados, mas esses valores podem ser usados para aplicações específicas.

Os valores de DLC estão representados na tabela abaixo.

Tabela 2 - Código de comprimento de dados para número de dados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Número de bytes de dados | DLC 0 | DLC 1 | DLC 2 | DLC 3 |
| 0 | D | D | D | D |
| 1 | D | D | D | R |
| 2 | D | D | R | D |
| 3 | D | D | R | R |
| 4 | D | R | D | D |
| 5 | D | R | D | R |
| 6 | D | R | R | D |
| 7 | D | R | R | R |
| 8-15 | R | D/R | D/R | D/R |

D: dominante, R: recessivo.

#### Dados

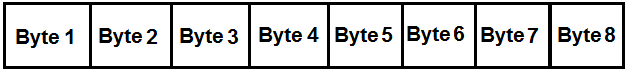


Figura 12 – campo de dados.

Fonte: Autoria própria.

A quantidade de dados em uma mensagem pode variar de zero a oito bytes, definidos no campo de controle. Uma mensagem com zero byte pode ser usada para indicar eventos, onde o evento é identificado pelo ID da mensagem.

#### CRC

O CRC contém código cíclico de redundância com 15 bits e no final um bit recessivo delimitador. A sequência de verificação da mensagem é derivada do código BCH. O código CRC contém uma distancia de Hamming de seis, podendo detectar até cinco bits errados distribuído nos campos de SOF, arbitragem, Controle e dados.

O CRC é gerado por um algoritmo de divisão polinomial. Onde os bits dos campos de SOF, arbitragem, controle e dados são divididos pelo polinômio gerados do CRC e o resto desta divisão é transmitido no campo de CRC junto à mensagem.

O seguinte polinômio gerador é usado pelo protocolo CAN:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | (1) |  |

No receptor a divisão pelo polinômio é feita incluindo o CRC e se o resto desta divisão não for zero existe um erro na transmissão e um frame de erro deve ser enviado pelo receptor para que a mensagem seja retransmitida.

O campo de CRC termina com o delimitador, sendo ele um bit sempre recessivo.

#### ACK

O campo de ACK é composto por dois bits, contendo o espaço de ACK e delimitador. O transmissor envia estes dois bits como recessivos e os receptores devem enviar no espaço de ACK um bit dominante se a transmissão foi recebida corretamente.

O delimitador deve ser um bit recessivo assim como o delimitador de CRC, sendo assim o espaço de ACK é cercado por bits recessivos. Se o transmissor detectar um bit dominante ele sabe que pelo menos um nó recebeu a transmissão corretamente.

#### EOF

O EOF, ou fim de frame, que delimita uma mensagem CAN é uma sequência de sete bits recessivos, onde o *bit-stuffing* está desabilitado, se um bit dominante for recebido durante o período de EOF um erro de *stuffing* será indicado.

### Tipos de mensagens

O protocolo CAN apresenta quatro tipos diferentes de mensagens: mensagens de condições normais de envio de dados, mensagens de dados (*Data Frame*), mensagens remotas (*Remote Frame*), e as mensagens de condições de erro: mensagem de erro (*Error Frame*) e mensagem de sobrecarga (*Overload Frame*).

#### Mensagem de dados

A mensagem de dados apresenta uma estrutura completa com todos os campos SOF, arbitragem, controle, dados, CRC, ACK e EOF. O campo de dados pode conter nenhum dado, sendo a mensagem usada para indicar eventos a outros controladores.

#### Mensagem remota

Um nó do barramento que precisa receber determinado dado para realizar ações, pode iniciar a transmissão da mensagem contendo este dado enviando uma mensagem remota.

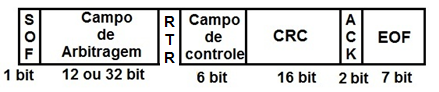


Figura 13 – A mensagem remota.

Fonte: Autoria própria.

A mensagem remota apresenta os campos da mensagem de dados excluindo o campo de dados, como apresentado na Figura 13, e contrário às mensagens de dados, o bit RTR deve ser recessivo nas mensagens remotas.

#### Mensagem de erro

Uma mensagem de erro é enviada por qualquer nó que detecta um erro no barramento. Os erros podem ocorrer por diferenças nos tempos de amostragem, diferenças nos limites de transição de bits, ou dispersão do sinal durante a transmissão. Os erros detectados por nós localmente são indicados pelo envio de uma sequência de seis bits de mesma polaridade seguidos, acarretando em um erro de stuffing e assim os outros nós irão indicar erro com mais seis bits de mesma polaridade.

A mensagem de erro consiste nos bits de erro, de seis a doze bits de mesma polaridade, e oito bits recessivos de delimitador de erro. Após o delimitador de erro e mais três bits o transmissor tenta reenviar a mensagem que foi corrompida.

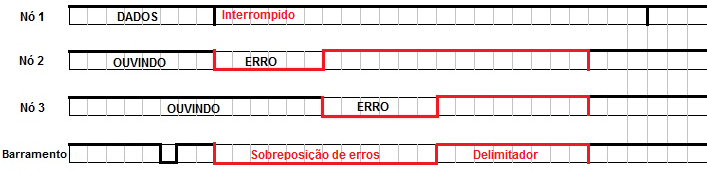


Figura 14 – condição de erro.

Fonte: Autoria própria.

A Figura 14 mostra um caso em que somente o nó dois reconheceu um erro na transmissão da mensagem, então ele inicia uma mensagem de erro, os outros nós reconhecem a sequência de seis bits de mesma polaridade e indicam erro também. Após o delimitador, oito bits recessivos, o nó um aguarda mais três tempos de bit e tenta reenviar a mensagem que foi corrompida.

Para prevenir que um nó defeituoso sobrecarregue o barramento com mensagens de erro, o envio de mensagens de erro pode ser restringido para este nó. Durante o tempo de delimitador de erro o nó pode verificar se ele foi o primeiro a indicar o erro e então um contador é incrementado e se este contador passa um valor definido de 128 ele entra no estado de erro passivo, onde ele só indica erro com bits recessivos, não interferindo no barramento. Se o contador passar o valor de 255 o nó é desconectado do barramento só podendo se reconectar em um *reset* por *software*.

#### Mensagem de Sobrecarga

Um nó do barramento que, devido a alguma condição interna, não possa receber mensagens por um tempo pode mandar uma mensagem de sobrecarga. Uma mensagem de sobrecarga pode ser mandada também, se um bit dominante for detectado em um dos dois primeiros bits durante o tempo entre mensagens, tempo de três bits necessários entre mensagens válidas.

A mensagem de sobrecarga é igual à mensagem de erro. A única diferença é que uma mensagem de sobrecarga não incrementa o contador de erro. Quando uma mensagem de sobrecarga é enviada os transmissores devem esperar para enviar a próxima mensagem.

# PADRÃO J1939

O protocolo J1939 foi desenvolvido pela associação de engenheiros automotivos, SAE, para ser aplicado em veículos comerciais. O protocolo é baseado no protocolo CAN e define como os dados são transmitidos entre unidades eletrônicas de controle (ECU) dentro da rede do veículo.

O J1939 apresenta algumas características particulares como: identificador de 29 bits, taxa de bits de 250kb/s ou 500kb/s, transmissões em broadcast ou ponto-a-ponto, protocolo de transporte de até 1785 bytes de dados, gerenciamento de rede e definição de grupo de parâmetros para veículos comerciais.

Existem diversos protocolos derivados do J1939, que divergem no grupo de parâmetros e na camada física, sendo elas: a ISO 11783, ou ISOBUS, desenvolvida para tratores e maquinários agrícolas e florestais; NMEA2000, para dispositivos eletrônicos marinhos; FMS, para sistemas de gerenciamento de frota.

## FORMATO DE MENSAGENS

O padrão J1939 define os grupos de parâmetros (PG), que são conjuntos de parâmetros pertencentes a um mesmo assunto e mesmas taxas de transmissão. As mensagens J1939 pertencem a algum PG, sendo ele apontado dentro do identificador da mensagem.

O identificador das mensagens do protocolo J1939 contém o número do grupo de parâmetros, prioridade, endereço de fonte, endereço de destino (quando mensagem ponto-a-ponto) e bits de página de dados.

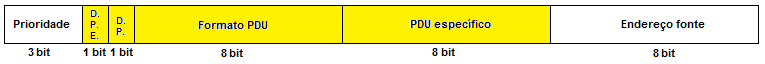


Figura 15 – Identificador J1939.

Fonte: Autoria própria.

A Figura 15 mostra a configuração de cada região do identificador das mensagens do protocolo J1939, as regiões marcadas em amarelo caracterizam o número do grupo de parâmetro (PGN).

O PGN é o identificador único de cada PG composto por formato de PDU, PDU específico, página de dados (DP) e extensão de página de dados (EDP). Um PGN pode ser do tipo global, onde os dados são enviados a todos os nós da rede (*Broadcast*), ou específicos, que podem ser enviados a um nó especifico da rede.

Em PGNs globais o formato PDU é um valor maior ou igual a 240 e o PDU específico é uma extensão do grupo. Para PGNs específicos o formato PDU é um valor menor ou igual a 239 e o PDU específico deve ser zero. Assim são possíveis 8672 PGs (2\*(240+(16\*256))=8672).

Quando uma mensagem ponto-a-ponto for enviada, o PDU específico deve ser o endereço de destino da mensagem, se o valor for definido em 255, sendo este o valor de endereço global, todos os ECUs devem ouvir e responder de acordo a esta mensagem.

Para cada parâmetro de um grupo de parâmetros será associado um número de parâmetro suspeito (SPN), para fins de identificação de controladores com operação anormal.

## PROTOCOLO DE TRANSPORTE

Quando mais de oito bytes de dados precisam ser enviados em um grupo de parâmetros, a comunicação dos dados é feita através de várias mensagens de dados. Esse tipo de PGN recebe o nome de *multi-packet*.

Se um PG, definido como capaz de múltiplos pacotes, precisa ser enviado com menos de oito bytes de dados, os dados podem ser enviados em uma mensagem, mas com o campo de DLC definido como oito. Quando mais de oito bytes precisam ser enviados, então a função de protocolo de transporte (TP) é usada.

Existem dois tipos de TP, mensagens de anuncio em broadcast (BAM) e por gerenciamento de conexão (CM).

Quando uma mensagem comprida precisa ser enviada em um mesmo PGN sem um destino específico ela é chamada de mensagem de broadcast. Para enviar uma mensagem de broadcast de múltiplos pacotes o transmissor precisa enviar uma mensagem de BAM. Esta mensagem é um aviso a todos os nós da rede de que uma mensagem de múltiplos pacotes será enviada.

A mensagem de BAM apresenta o PGN da mensagem, o tamanho total, e o número de pacotes em que a mensagem foi separada. Os nós interessados irão alocar os recursos necessários para receber e remontar a mensagem.

Mensagens longas que precisam ser enviadas a nós específicos devem utilizar o protocolo de gerenciamento de conexões. A comunicação deve ser iniciada com uma mensagem de CM onde o byte de controle indica que a mensagem está pronta para envio, *ready to send (RTS)*. O receptor deve responder com uma mensagem de CM onde o byte de controle indica que o nó está pronto para receber, *clear to send* (CTS). O transmissor então envia uma porção dos dados indicado na mensagem de CTS, através de uma mensagem de transferência de dados (DT). A sequência de mensagens de CTS e DT continua até que todos os dados tenham sido transmitidos. Ao final da transmissão dos dados o receptor envia uma mensagem de fim de CM com o byte de controle indicando o reconhecimento de fim de mensagem (EOM).

## ENDEREÇAMENTO

No protocolo J1939 todos os nós, ou controles eletrônicos na rede de um veículo devem conter um endereço único para que as comunicações sejam realizadas. Em sua maior parte, os nós já contém um endereço pré-definido internamente e já o utilizam no início das comunicações.

O protocolo ainda prevê um método de designar endereços dinamicamente para dispositivos ou funções que sejam adicionadas futuramente. Existem dois métodos para realizar isto.

No primeiro, o novo dispositivo sem endereço deve mandar uma mensagem de *Address Claim* informando aos outros nós qual endereço ele gostaria de tomar. Todos os outros dispositivos da rede devem verificar se já possuem o endereço informado em suas tabelas de endereços. Se um dispositivo mais prioritário já contém o endereço, ele deve informar o conflito enviando uma mensagem de *Address Claim.*

No segundo, o novo dispositivo deve requisitar aos outros nós seus endereços e então adotar um endereço que não seja igual a algum dos outros.

# APLICAÇÕES DO PROTOCOLO CAN

Nas páginas seguintes serão demonstradas algumas aplicações do protocolo CAN, as quais existem em diversas áreas, como aeroespacial, agricultura, automação, automobilística, construção e marítimo. Essas aplicações mostram a importância e abrangência desse método de comunicação, e a necessidade de um maior conhecimento sobre o mesmo. O seu sucesso deve-se a sua confiabilidade, robustez e baixo custo de implementação.

## AUTOMAÇÃO

A utilização do barramento CAN nas indústrias é um pouco mais recente do que em outras áreas. Em 1992 foi criado o grupo internacional de fabricantes e usuários de CAN na automação, CiA. O grupo criou um padrão de alto nível baseado em CAN, em 1994, chamado hoje de CANopen.

CANopen implementa as camadas do modelo OSI a partir da camada de rede e outras, mais altas, até a camada de aplicação. O padrão implementa protocolos de gerenciamento de rede, de serviço de dados, de processamento de dados, sincronização, temporização e emergência.

O barramento CAN foi altamente aceito no ambiente industrial por se tratar de um barramento robusto, possuir alta rejeição a ruídos, utilizar par trançado blindado, e ter baixo custo de implementação.

## AUTOMOTIVA

O protocolo CAN apresenta grande influência no projeto dos veículos automotivos, pois o protocolo foi desenvolvido inicialmente para que fossem diminuídos o peso e a quantidade de cabos no interior do veículo. Com o passar do tempo diversos equipamentos e subsistemas foram incluídos nos veículos e todos eles foram interligados através do barramento CAN.

Devido ao grande número de controladores nos veículos atuais, eles apresentam mais de um barramento, com diferentes velocidades. Por exemplo, o barramento do motor deve ter uma taxa de transmissão elevada para que as mensagens trafeguem o mais rápido possível. A imagem a seguir ilustra os barramentos CAN de um veículo, cada linha de cor diferente representa um barramento CAN diferente.

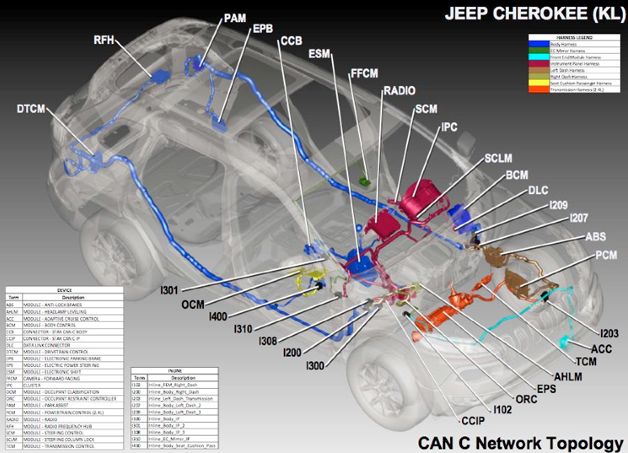


Figura 16 – ECUs e barramentos em um veículo.

Fonte: <http://www.autoblog.com/2014/09/12/risk-cyber-threats-may-increase-cars-computers-active-controls/>

## NÁUTICA

Barcos e navios também contêm muitos controladores e sistemas interligados, que necessitam de uma rede unificada para que possam interagir. Foi criado um padrão de comunicação de rede para os controladores de barcos, baseado no padrão J1939 de máquinas pesadas, o NMEA 2000. Esse padrão permite uma taxa de transmissão de 250 kbps e habilita diversos componentes com o NMEA a se comunicarem. Os principais são o GPS, piloto automático, instrumentação de vento, instrumentos de navegação, como mostra a figura abaixo.

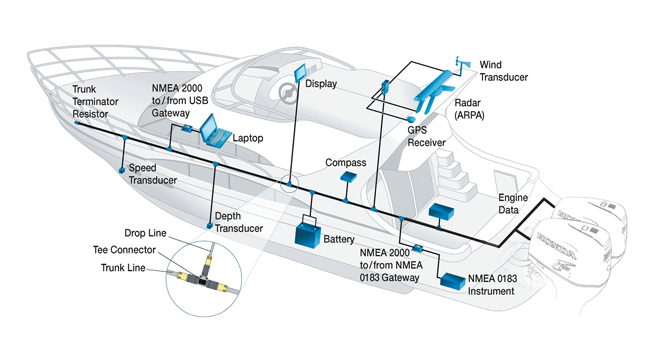


Figura 17 – Barramento típico NMEA

Fonte: http://www.gommonauti.it/ptopic55590\_honda\_bf200\_e\_cavi\_nmea.html&start=20

## AGRÍCOLA

O maquinário agrícola pesado também precisa de controladores eletrônicos para funcionar de forma eficiente e segura. Vários controladores e sensores eletrônicos precisam ser interligados nas máquinas. Para isso foi desenvolvido o padrão para máquinas agrícolas ISO 11783, ou ISOBUS.

Além de ter diversos controladores internos na máquina, ela ainda deve ter a funcionalidade de se comunicar e controlar implementos externos produzidos por fabricantes diferentes. Sem um padrão de comunicação entre máquina e implementos seria necessária uma massa de cabos para os controles de cada implemento, forçando o usuário a amontoar controladores dentro da máquina ou a se fidelizar a uma marca específica.

O padrão ISOBUS é baseado no protocolo J1939, definindo algumas particularidades da camada física e definindo alguns PGNs específicos para máquinas agrícolas. A imagem a seguir mostra um display, que pode controlar diversos tipos de implementos através do padrão ISOBUS, além de poder monitorar as funcionalidades da máquina.



Figura 18 – Display conectado ao barramento CAN.

Fonte: CNH industrial.

No modelo de tratores T7 da New Holland, o emprego do barramento CAN diminuiu o uso de cabos em vinte por cento e o número total de conectores elétrico pela metade, e também possibilitou o uso de monitores de diagnóstico e controle de implementos (CNHi, 2014).



Figura 19 – Controles ligados ao barramento CAN.

Fonte: CNH industrial

As máquinas agrícolas que utilizam o barramento CAN podem se beneficiar deste barramento para controlar diversas funções da máquina. Algumas colheitadeiras utilizam o barramento para controlar diversas funções da máquina, como o sistema de colheita, a interação com o operador pelo display e joystick, a orientação manual e por GPS (direção e velocidade) e a comunicação entre o motor e o controlador da transmissão. A imagem abaixo mostra a colhedora de cana de açúcar, que utiliza em grande parte de seu sistema o barramento CAN para controle de diversas funcionalidades da máquina.



Figura 20 – Colhedora de cana de açúcar.

Fonte: CNHi.

Mesmo com a robustez e confiabilidade do barramento CAN, as máquinas agrícolas precisam da garantia do funcionamento correto do barramento, por se tratarem de máquinas pesadas e perigosas. Várias análises podem ser feitas no barramento para garantir o seu funcionamento correto, dentre elas estão um conjunto de formulas para verificar se as mensagens são recebidas em tempo hábil.

Nos próximos capítulos serão apresentadas as formas de análise do barramento para garantir que o barramento funcione corretamente.

# ATRASO DO TRÁFEGO DE MENSAGENS NO BARRAMENTO CAN

Esta seção consiste na descrição de noções e variáveis fundamentais que representam o comportamento do protocolo CAN na análise do tempo de resposta. A análise é necessária devido ao fato de que somente uma mensagem vence o processo de arbitragem (presente na seção 2.5.1) a cada vez, gerando atrasos nas demais mensagens. Através desta seção, introduz-se o leitor às seções subsequentes que realizam a análise de fato, como as equações determinísticas (seções 6.1 e 6.2) e a análise de probabilidade (seção 6.4).

No momento em que dois ou mais nós (módulos) queiram ocupar o barramento para enviar suas mensagens, o mecanismo de arbitragem presente durante o início das mensagens define quem tomará o barramento e enviará a sua mensagem. Isto faz com que somente um dos nós transmita, enquanto os outros esperem essa transmissão.

Por exemplo, se em *t=0* os nós A, B, C querem cada um enviar uma mensagem com prioridades respectivas de 1,2 e 3,a transmissão no barramento se dará da seguinte forma:

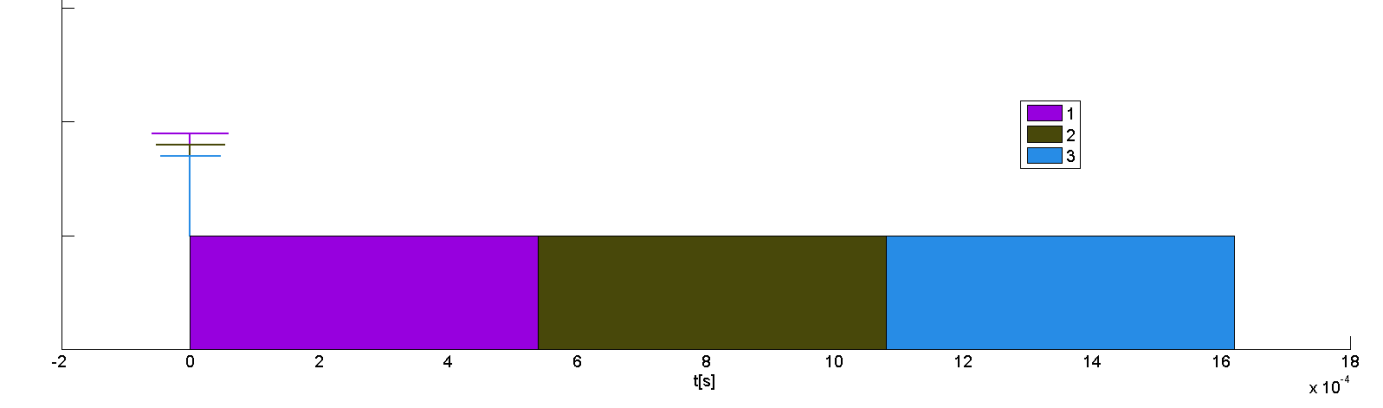


Figura 21 - Exemplo de arbitragem com 3 mensagens

Fonte: Autoria Própria

A mensagem que primeiro ocupa o barramento é a com prioridade igual a 1, pois ela tem a maior prioridade entre as mensagens que estão disputando pelo barramento. Isso ocorre pois quanto menor o valor numérico da prioridade, mais prioritária é a mensagem, fazendo com que a prioridade 1 seja maior que a 2, que por ventura é maior que a 3.

## TEMPO DE TRANSMISSÃO

Uma mensagem com prioridade *k* ocupa o barramento durante o seu *tempo de transmissão*, *Ck*, correspondendo ao produto entre o número de bits a serem transmitidos, *Nk*, e o tempo de bit, *tb*.

O tempo de bit é o período de tempo em que um bit é mantido no barramento, sendo numericamente igual ao inverso da frequência de barramento, e é igual em todas as mensagens e módulos.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

O valor do tempo de transmissão de uma mensagem com prioridade k é importante, pois é o tempo em que k ocupa o barramento após ganhar a arbitragem, impossibilitando a transmissão de outras mensagens.

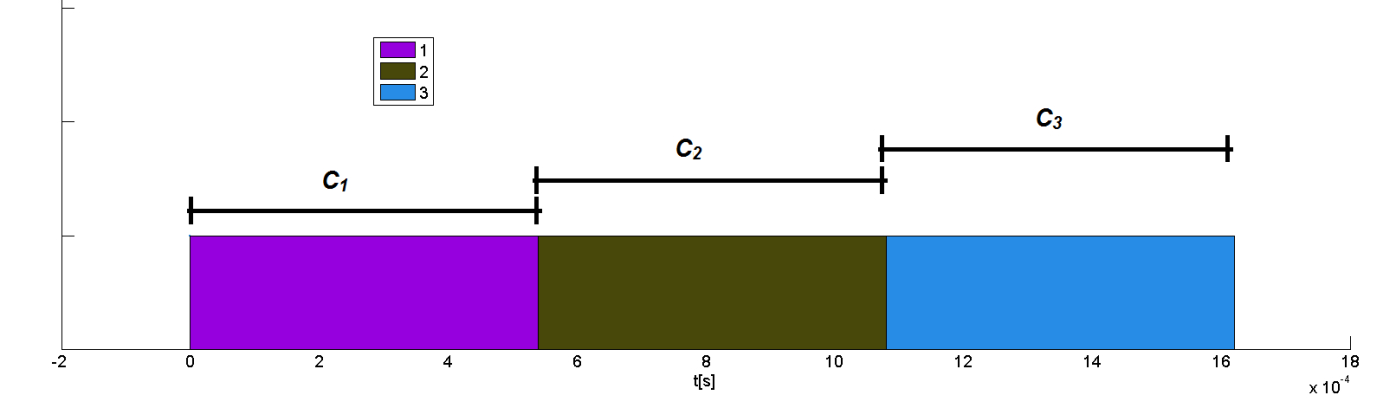


Figura 22 - Tempos de Transmissão

Fonte: Autoria Própria

## TEMPOS DE INÍCIO E FIM DE TRANSMISSÃO

O início de uma transmissão ocorre quando uma mensagem com prioridade *k* que espera pela sua vez no barramento acaba ganhando o processo de arbitragem e inicia a transmissão de seus dados, assim, o nome dado a esse tempo é *tempo de início de transmissão, Sk*.

Uma vez que a transmissão foi iniciadaa mensagem terá sido completamente enviada após um tempo *Ck* (*tempo de transmissão*), , livrando o barramento para uma nova mensagem e iniciando uma nova arbitragem (caso existam mensagens pendentes). O nome dado ao momento em que a mensagem com prioridade *i* cessou de existir no barramento é *tempo de fim de transmissão*, *Ek*.

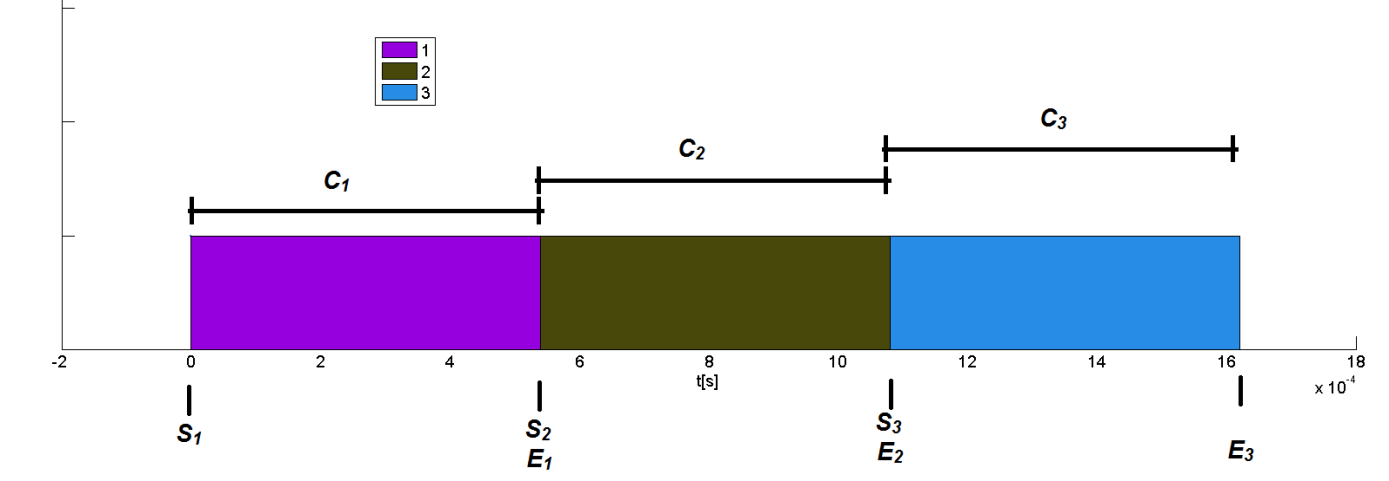


Figura 23 - Transmissão com indicação de início e fim

Fonte: Autoria Própria

De acordo com a definição dada, então:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

## TEMPO DE INVOCAÇÃO

Para que uma mensagem seja transmitida, ela precisa participar e vencer no processo de arbitragem durante o início de cada frame CAN.

Assim, denomina-se *tempo de invocação* de uma mensagem k, *ik*, o momento em que uma mensagem está pronta para participar do processo de arbitragem, independente do fato se a arbitragem ainda não foi iniciada ou se outra mensagem está ocupando o barramento.

O *tempo de invocação* é tomado como o momento em que o módulo deseja enviar a mensagem no barramento, isso sendo possível ou não. Ele difere em relação ao *tempo de início de transmissão*, pois não se constitui do momento em que a mensagem de fato é transmitida no barramento, mas quando o módulo começa a tentar vencer a arbitragem para que ela seja transmitida.

Isso pode ser ilustrado na Figura 24 onde o tempo de invocação da mensagem 2, *i2*, é de 0.7ms, fazendo com que, diferentemente do caso anterior, as mensagens 2 e 3 trocassem de lugar.

Nas imagens subsequentes os tempos de invocação serão ilustrados por um ‘T’ presente acima das mensagens e com cores iguais as mesmas.

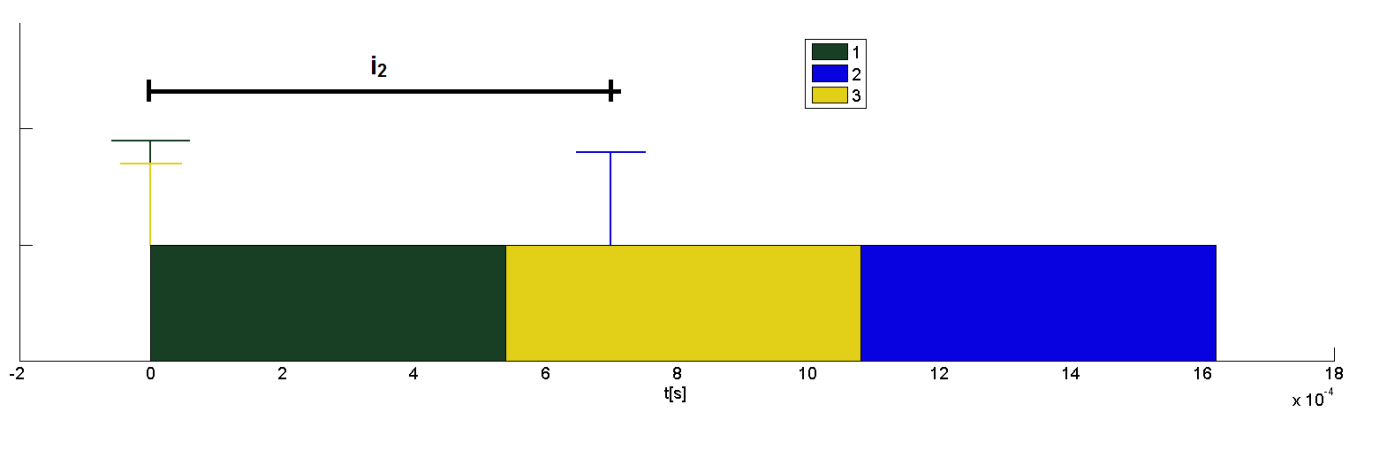


Figura 24 - Transmissão com indicação do tempo de invocação da mensagem 2

Fonte: Autoria Própria

Se o barramento estiver desocupado quando uma única mensagem for invocada, existe a garantia de que ela será transmitida, uma vez que só uma mensagem participa da arbitragem, assim como está ilustrado no exemplo abaixo:

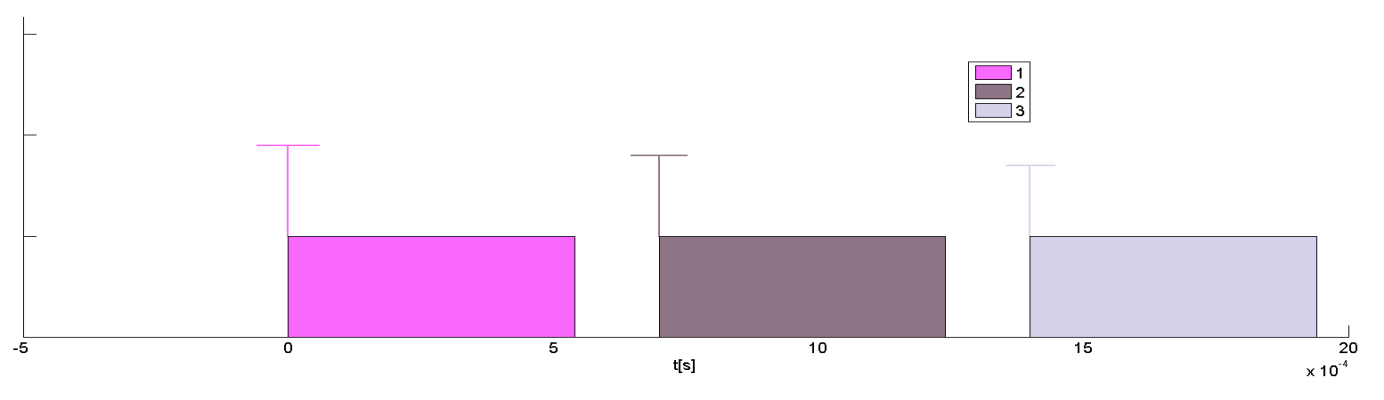


Figura 25 - Tempos de invocação coincidiram com o de inicio de transmissão

Fonte: Autoria Própria

## INSTÂNCIA DE UMA MENSAGEM

Até o momento, as figuras apresentadas consistiram em uma única transmissão de cada mensagem, o que dificilmente corresponde a um caso prático, pois as mensagens são enviadas mais de uma vez, periodicamente ou não, gerando várias execuções.

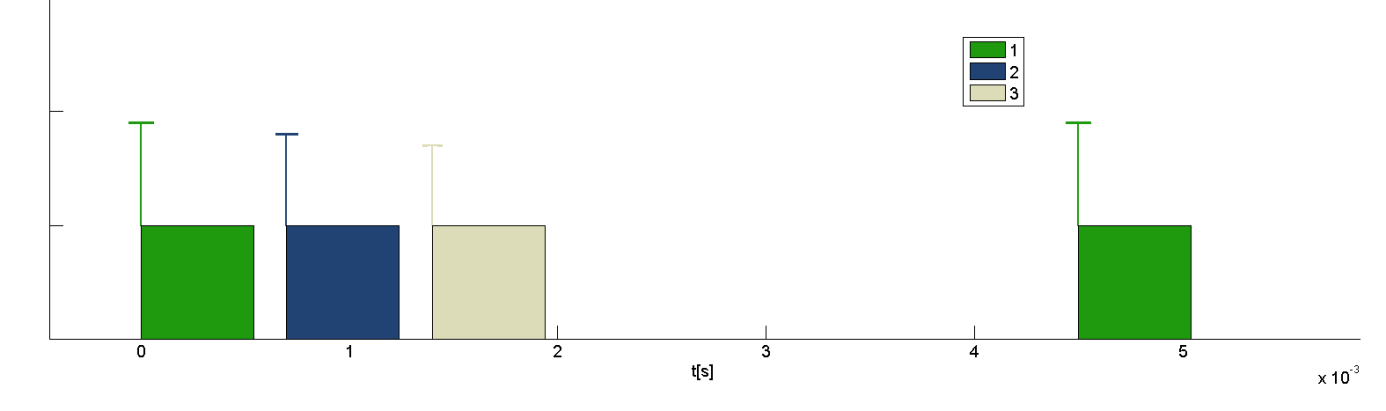


Figura 26 - Duas instâncias da mensagem 1

Fonte: Autoria Própria

Utilizando a figura acima como exemplo, temos três mensagens diferentes, cada uma com sua prioridade (1, 2 e 3). As transmissões em *t=0* e *t=4.5ms* são a mesma mensagem pois o seu identificador é o mesmo, independente dos dados que cada uma carrega.

A mensagem com prioridade 1 é transmitida no barramento duas vezes, uma em *t=0* e outra em *t=4.5ms*, levando a duas *instâncias* dessa mensagem e a somente uma *instância* das demais.

Cada instância tem o seu próprio *tempo de transmissão*, *tempo de início de transmissão*, *tempo de fim de transmissão* e *tempo de invocação*, o que nos leva a expandir essas variáveis (3), adicionando o subscrito de cada instância *j* para cada mensagem *k*.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |
|  |  | (5) |

Assim, observando a figura, pode-se construir uma tabela com os tempos iniciais e finais de transmissão para cada mensagem *k* e instância *j*:

Tabela 3 - Tempos iniciais e finais de transmissão para cada mensagem *k*  e instância *j*

|  |  |  |  |
| --- | --- | --- | --- |
| , | k=1 | k=2 | k=3 |
| j=1 | {0, 540us} | {700us, 1.24ms} | {1.4ms, 1.94ms} |
| j=2 | {4.5ms, 5.04ms} | --- | --- |

## TEMPO DE RESPOSTA

Nos tópicos anteriores foi possível observar que, devido à ocupação do barramento por outras mensagens, o início de transmissão de uma instância de mensagem nem sempre ocorre exatamente no tempo em que a mensagem e o módulo estão prontos para transmissão (*tempo de invocação*). Isto insere um tempo de espera, *q*, anterior à execução da referida instância.

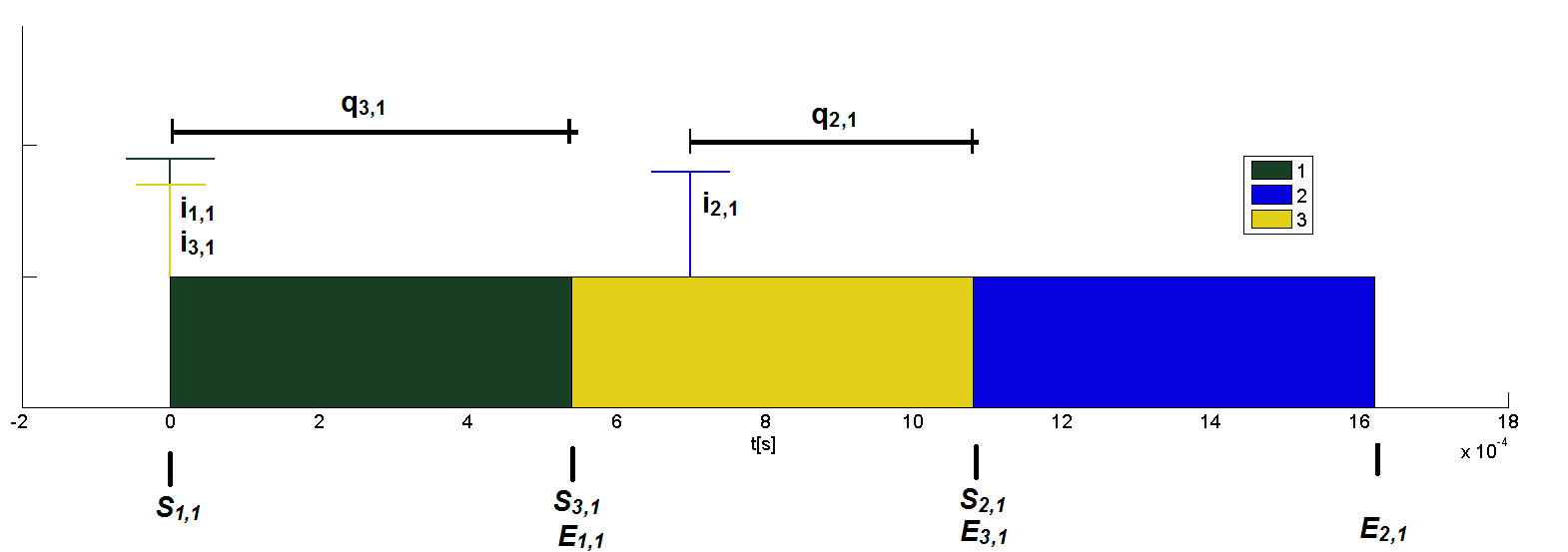


Figura 27 - Mensagem 2 com invocação e tempo de espera diferente de zero

Fonte: Autoria Própria

A partir da definição do *tempo de espera*, e reutilizando a noção de instância e mensagem, obtém-se que:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Obtém-se também que o *tempo de resposta, rk,j*, definido como o tempo em que a mensagem levou desde que o módulo e a mensagem estão prontos para transmissão (*tempo de invocação*) até o fim de transmissão (), é igual a:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Utilizando as relações (6) e (4)

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

O *tempo de resposta* pode ser entendido como o tempo que existe entre o desejo do módulo transmissor de ocupar o barramento para enviar a sua mensagem e o módulo receptor receber a mensagem completa.

Através desse ponto de vista, nota-se que é importante mensurar o *tempo de resposta*, uma vez que atrasos não previstos podem acarretar em problemas de comunicação entre os módulos, ocasionando problemas no sistema como um todo.

Naturalmente, a equação (8) mostra que o tempo de resposta depende do tempo em que a mensagem esperou para ganhar a arbitragem do barramento, o que leva a conclusão de que o tempo de resposta muda entre cada instância da mensagem, fazendo-se necessária uma análise mais minuciosa dos seus limites.

## MENSAGENS PERIÓDICAS

Muitas das mensagens que trafegam em um barramento CAN são ditas periódicas, pois elas são invocadas a cada tempo *T* após a sua última invocação.

A utilização dessa estratégia tem como uma de várias vantagens a melhor utilização do barramento: algumas mensagens não têm a necessidade de ser enviadas sempre que os seus dados estiverem prontos, ou porque o módulo receptor pode não conseguir processá-las tão rapidamente ou porque o dado que está sendo enviado não sofre rápidas variações.

Um exemplo dessa situação é quando uma envia o estado da porta do veículo (aberta ou fechada). Por mais que seja um dado importante, a porta do veículo não sofre rápidas variações (período muito maior que milissegundos) e as medidas a serem tomadas ao detectar esse evento não sofrerão grandes impactos se a transmissão ocorrer a cada 1*ms* (tempo mínimo determinado por um *ADC* hipotético do sistema) ou 50 *ms*, assim, para não ocupar excessivamente o barramento, pode-se criar um limite inferior para o tempo entre invocações da mensagem.

Outra vantagem para a utilização de mensagens periódicas é aumentar a previsibilidade do sistema, pois assim podem-se detectar erros dos módulos e mensurar limites de atraso para a operação de cada sistema.

A detecção de erro pode ser implementada quando o módulo receptor não recebe a mensagem por mais de alguns períodos *T* da mensagem, significando que o transmissor teve algum problema ou que o barramento foi danificado, possibilitando a tomada de medidas corretivas automáticas.

Os limites de atraso máximos para a operação dos sistemas são utilizados como base para o cálculo dos períodos de invocação (levando em conta uma análise dos tempos de resposta), diminuindo a utilização excessiva do barramento e, consequentemente melhorando os valores de tempo de resposta para todas as mensagens.

### Distância entre instâncias ou invocações

A distância, *dj,k*, entre duas instâncias, ou invocações, *j* e *k*, é definida pela quantidade de instâncias, ou invocações, que existem entre elas mais um.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Por exemplo, a distância entre duas invocações sucessivas (*j* e *j-1*) é sempre igual a um, pois elas não contêm invocações entre elas mesmas.

### Período

O período consiste no intervalo *Tk* fixo entre duas invocações sucessivas (*j* e *j-1*, *dj,j-1=1*) da mesma mensagem *k*, equacionado em (10) e (11) e exemplificado na figura para *k=1*.

|  |  |  |
| --- | --- | --- |
|  |  | (10) |
|  |  | (11) |

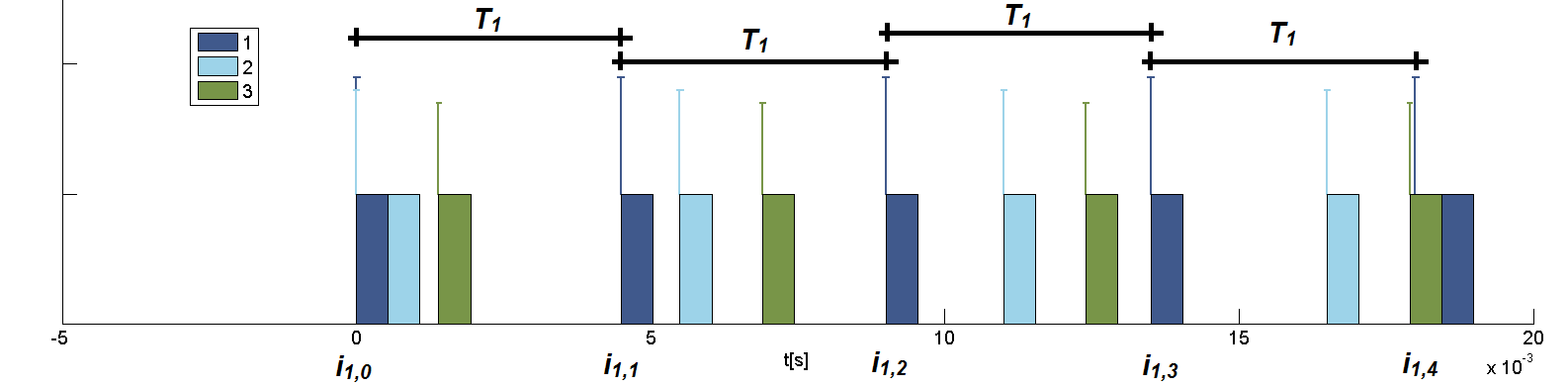


Figura 28 - Transmissão com indicação de período entre instâncias da mensagem 1

Fonte: Autoria Própria

Somando (10) e (11) temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

A equação (12) fornece o tempo de invocação entre invocações com *d=2*, assim, expandindo o conceito para uma instância *n* e outra *g*:

|  |  |  |
| --- | --- | --- |
|  |  | (13) |
|  |  | (14) |

Assim, sabendo-se o período *Tk* de uma mensagem, e uma de suas invocações , pode-se estimar o restante das invocações através da equação (14).

Por exemplo, se *T1 é 1s* e *i1,3=3,43s* (*g=3*), tem-se os seguintes valores:

Tabela 4 - Tempos estimados de invocação com base na terceira instância

|  |  |
| --- | --- |
| N | (s) |
| 0 | 0.43 |
| 1 | 1.43 |
| 2 | 2.43 |
| 3 | 3.43 |
| 4 | 4.43 |
| 5 | 5.43 |

# ANÁLISE DO TEMPO DE RESPOSTA

A análise do tempo de resposta das mensagens CAN tem como principal motivação obter conhecimento sobre o comportamento do sistema, indicando limitações e características do mesmo, o que torna possível a análise dos impactos finais no sistema veicular como um todo.

A próxima seção (6.1) consiste em uma revisão bibliográfica sobre a análise de pior caso de tempo de resposta no escalonamento de sistemas operacionais preemptivos e com prioridade fixa, considerando *jitter* e tarefas esporadicamente periódicas.

Subsequentemente (seção 6.2), explora-se a literatura que aborda equações oriundas da análise de sistemas preemptivos, as quais são expandidas e aplicadas a não-preemptivos, o que leva à sua utilização no barramento CAN.

A seção 6.3 apresenta conclusões acerca da utilização da abordagem da bibliografia, levando à uma preferência por uma abordagem probabilística, proposta e desenvolvida na seção seguinte.

Os impactos podem ser dos mais variados, uma vez que existem várias ECUs, com diversas possíveis mensagens a serem enviadas e recebidas, cada uma com o seu *deadline*. Dependendo do impacto, e da quantidade de vezes em que o estouro de *deadline* ocorre (probabilidade), o mesmo pode ser aceitável quando se compara o efeito com o gasto de recurso necessário para incrementar a capacidade da rede de comunicação.

Adquirir o conhecimento de que a rede CAN atende aos requisitos de tempo, ajuda na depuração de outros tipos de problemas de comunicação entre os módulos, onde o tempo de resposta acaba por tornar-se uma possível causa de problema e dificulta o diagnóstico correto do funcionamento dos módulos.

Ao não receber uma mensagem esperada, módulos podem interpretar que o módulo transmissor foi desligado do barramento ou sofreu algum tipo de erro, tomando medidas de precaução, mas na verdade o sistema estava funcionando corretamente, e o problema foi no atraso ocasionado na mensagem pelo barramento.

Por esses motivos, introduz-se também a necessidade da análise do tempo de resposta quando se considera a arquitetura de um sistema. Ao mesmo tempo em que se busca segurança e confiabilidade, deseja-se também eficiência, evitando o superdimensionamento. Este pode até mesmo implicar na implantação de um barramento CAN extra.

Sem a análise do tempo de resposta é impossível mensurar a qualidade com que um barramento CAN pode atender o deadline das mensagens.

## ESCALONAMENTO DE SISTEMAS OPERACIONAIS

O primeiro trabalho publicado sobre a análise do tempo de resposta da rede CAN, também chamado de escalonamento da rede CAN, foi o de Tindell e Burns (1994). Nele, testes de escalonabilidade utilizados em sistemas operacionais preemptivos de prioridade fixa são adaptados com pequenas mudanças ao protocolo CAN, resultando em um conjunto de equações que fornece o pior caso de tempo de resposta para cada mensagem.

Esta seção terá como objetivo introduzir o leitor às equações presentes no artigo de Tindell (1992) que melhor se relacionam com a análise feita posteriormente por ele mesmo para o barramento CAN.

Tópicos apresentados por Tindell, como o atraso representado pelo escalonador e o método ótimo de ordenamento, não serão abordados aqui, pois são conceitos inexistentes no barramento CAN.

### Relação com sistemas operacionais de prioridade fixa

Em trabalhos prévios, Tindell (1992) partiu da equação (15) deduzida por Joseph e Pandya (1986). Ela fornece o pior tempo de resposta, *ri*, para uma determinada tarefa i, assumindo que as tarefas mais prioritárias que estaforam liberadas ao mesmo momento que *i*, denominado *momento crítico para o nível i*, e provado por Liu e Layland (1973), que corresponde à situação de pior caso de tempo de resposta para *i*.

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

O tempo de resposta é influenciado pelo tempo de execução de cada tarefa mais prioritária,, e pelo próprio tempo de execução de *i*, , pois o tempo de resposta é considerado desde o momento em que *i* está pronta para ser executada até o momento em que terminou de sê-lo.

Este tipo de análise baseia-se fortemente no conceito de tarefas periódicas, que têm grande uso em vários sistemas operacionais. Muitas vezes os seus períodos são utilizados como base para se estabelecer um tempo máximo de resposta ao qual o sistema deve obedecer.

A periodicidade de uma tarefa consiste em que a diferença de tempo (período ) entre a tarefa estar pronta para execução é sempre o mesmo (caso ideal) e determinado pelo arquiteto do sistema, ajudando na previsibilidade de todo o conjunto.

O somatório apresentado na equação (15) consiste na soma das interferências que as tarefas de prioridade maior que *i* geram nesta, ou seja, o pior caso de atraso considerando a quantidade máxima de vezes que a tarefa *i* pode ser preemptada por outras de maior prioridade.

Tabela 5 - Exemplo de conjunto de tarefas periódicas em um SO

|  |  |  |
| --- | --- | --- |
| *T(ms)* | *C (ms)* | prioridade |
| 2.5 | 1 | 1 |
| 3.5 | 1 | 2 |
| 6.0 | 1 | 3 |

Utilizando o conjunto fornecido pela Tabela 5, pode-se verificar o pior tempo de resposta da mensagem 3 através do escalonamento do momento crítico apresentado na Figura 29, de acordo com Liu e Layland (1973).

Nesta figura pode-se observar que a preempção da tarefa 3 em *t=2.5ms*  aumentou o seu tempo de resposta, somente voltando a execução em *t=4.5ms*, onde ainda faltava *0.5ms* para a completa execução da primeira instância da tarefa 3.

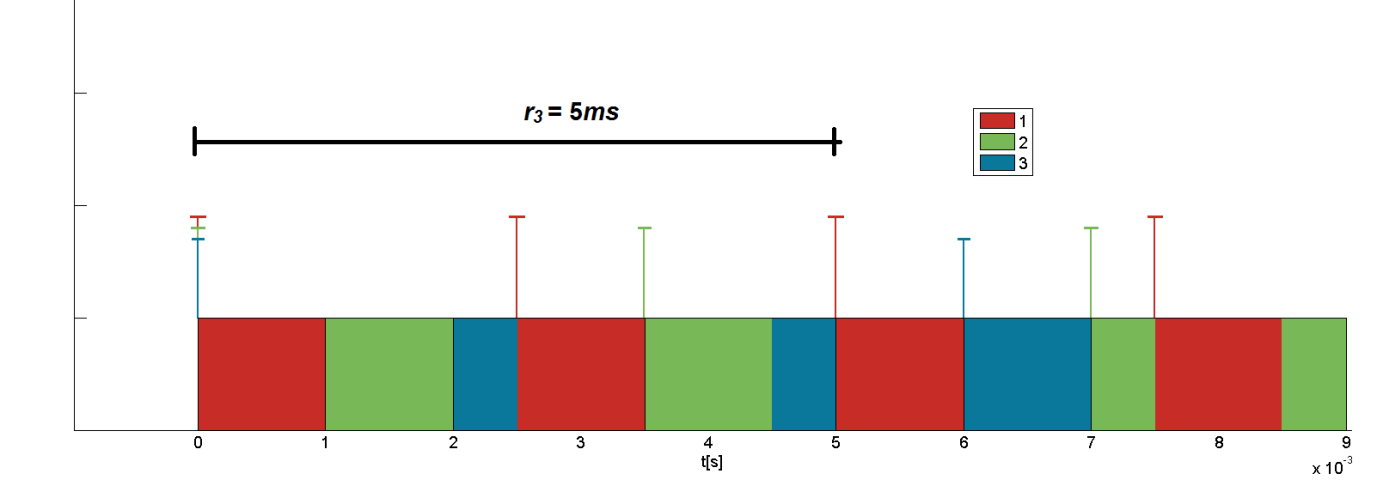


Figura 29 - Pior tempo de resposta da mensagem 3 em um sistema preemptivo

Fonte: Autoria Própria

Supondo que a prioridade de uma tarefa *j* é maior que a de *i* (, cada elemento de interferência da tarefa *j* na tarefa *i*, , consiste no tempo em que *j* ocupou o processador. O sistema analisado é preemptivo e de prioridade fixa, o que leva ao fato de que a tarefa *j*, e mais prioritárias que esta,sempre interrompem a tarefa *i*. Em outras palavras, a tarefa *i* nunca retorna à execução enquanto existe uma tarefa mais prioritária a ser executada, portanto, as tarefas mais prioritárias sempre interrompem uma tarefa *i* por um tempo múltiplo ao seu tempo de execução (este que corresponde a para uma tarefa *j*). Cada execução de outra tarefa que não seja a analisada *i* corresponde à um elemento de interferência.

O número de vezes em que uma tarefa mais prioritáriainterfere em *i* é sempre inteiro, devido ao fato de que uma tarefa menos prioritária conclui a sua execução somente quando nenhuma outra tarefa mais prioritária está pronta para execução. Isto garante que as tarefas mais prioritárias tenham executado por inteiro, uma ou mais vezes.

A interferência de uma tarefa *j* em outra *i* é denotada matematicamente por (16), onde existe uma interferência somente se *j* for mais prioritária.

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

O termo *k* é um valor inteiro como o maior número de instâncias de *j* que interferiram em *i*, e é o tempo de execução de uma mensagem *j*.

Para que se obtenha o *valor* de *k*, devemos avaliar quantas instâncias de *j* foram invocadas até o fim de execução da tarefa *i* (tempo de resposta de *i*, ).

Assim, divide-se o tempo final de *i* pelo período de invocação de *j*, levando equação (17), onde o operador significa a função de arredondamento para cima ().

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

Somando todas as interferências provenientes de tarefas mais prioritárias até um momento *t,* encontramos quanto tempo as tarefas mais prioritárias invocadas até *t* ocuparam o processador, conforme a equação (18).

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

Assim, desejando-se descobrir o tempo *ri*  em que a tarefa *i* completou a sua execução, e sabendo que ela sofreu uma interferência *K* de mensagens mais prioritárias até o instante *ri* e que ela executou uma vez até *ri*, obtemos a equação original (15) a partir da (19):

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

Na figura abaixo se pode observar novamente o escalonamento da Tabela 5. Ressalta-se a interferência (*Int1->3* e *Int2->3*) que as tarefas 1 e 2 causaram na tarefa 3, cada uma executando duas vezes até *r3* (*k2=k1=2)*, sendo que em *t=2.5ms* a tarefa 3 é preemptada pela tarefa 1, executando novamente em *t=4.5ms*.

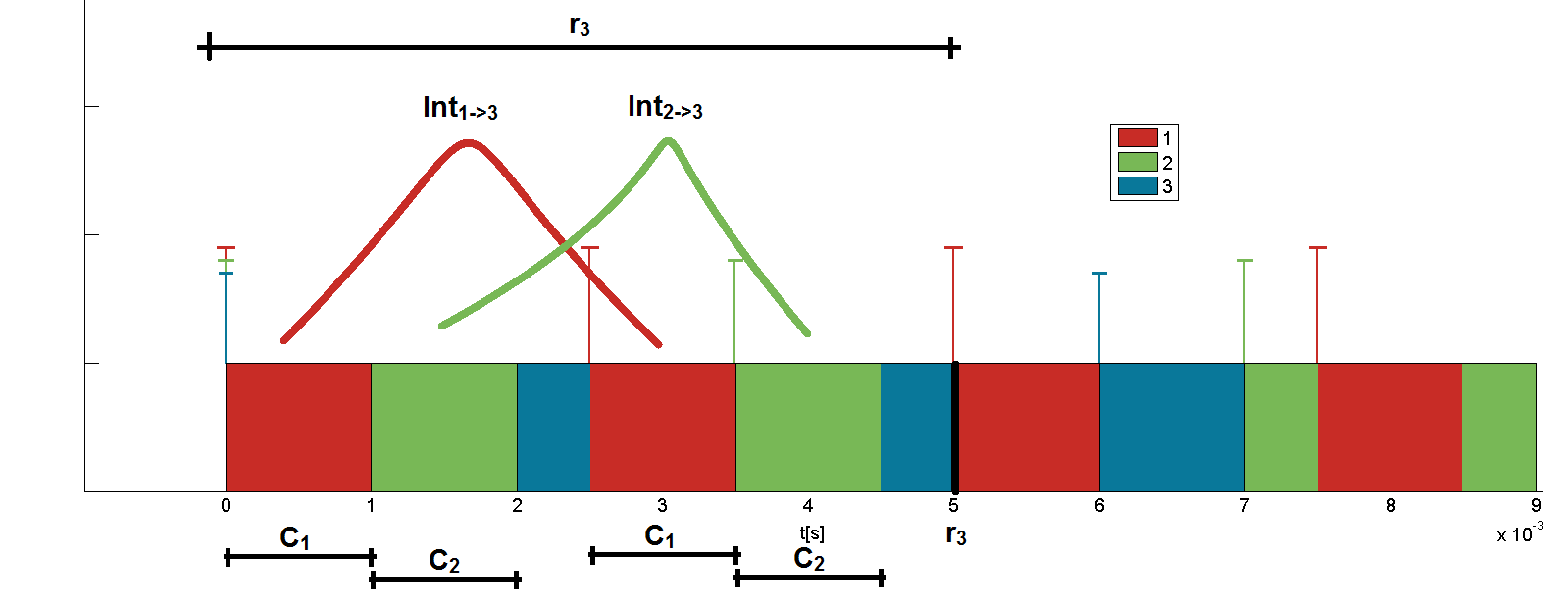


Figura 30 - Demonstração da interferência causada na mensagem 3 por outras mais prioritárias (1 e 2)

Fonte: Autoria Própria

Para resolver a equação (4), Tindell propõe um método iterativo (20), onde existe a convergência ( de somente se a ocupação total de barramento pelas tarefas mais prioritárias que *i* for menor que 1.

A ocupação do barramento por uma tarefa periódica *i* é dada pela razão entre o seu tempo de execução e o seu período.

|  |  |  |
| --- | --- | --- |
|  |  | (20) |
|  |  | (21) |

Além dessa limitação, a equação (20) somente fornece valores válidos se o tempo de resposta for menor que o período da tarefa, uma vez que ela não considera invocações anteriores da tarefa *i*. Isso fez com que Tindell desenvolvesse equações que avaliassem o tempo de resposta para um *deadline* arbitrário (maior que ).

Uma análise mais abrangente precisa ser realizada para um *deadline* arbitrário, pois se no momento em que uma tarefa *i* estiver sendo executada, uma nova instância da mesma for invocada (), uma interferência da primeira instância ocorrerá nessa nova invocação.

### Exemplo de aplicação da equação de Joseph e Pandya

Utilizando a equação (15) deduzida por Joseph e Pandya (1986), para avaliar o tempo de resposta da mensagem 3, inserida em um sistema preemptivo com o conjunto da Tabela 5, temos que:

O resultado é válido, pois é menor que o período da mensagem (*T3=6ms*) e é igual ao obtido através do desenho do escalonamento do sistema (Figura 29).

### Conceito de busy-period

Para a generalização das equações anteriores às tarefas com *deadline* arbitrário, Tindell (1992) utilizou o conceito de *busy-period* (período ocupado), introduzido por Lehoczky (1990).

O conceito de “período ocupado nível *i*” (*level i busy-period*) consiste em um espaço contíguo de tempo em que somente tarefas com prioridades iguais ou maiores que *i* utilizam o processador, impedindo a execução de tarefas menos prioritárias que *i*.

Lehoczky prova que o pior tempo de resposta de uma tarefa *i* está no período ocupado nível *i* criado pelo momento crítico para o mesmo nível; ou seja, se *i* e todas as tarefas mais prioritárias forem invocadas em , o pior tempo de resposta de *i* estará em entre e o fim do período ocupado, que é o momento em que o processador fique desocupado ou uma tarefa de prioridade menor que *i* seja executada.

Tabela 6 - Conjunto de tarefas periódicas em um SO onde a tarefa 3 tem um tempo de resposta maior que o seu período

|  |  |  |
| --- | --- | --- |
| *T(ms)* | *C (us)* | prioridade |
| 2.4 | 1000 | 1 |
| 2.4 | 1000 | 2 |
| 4.2 | 680 | 3 |

Na figura abaixo está representado o período ocupado nível 3 de momento crítico da Tabela 6, mostrando qual tarefa está utilizando o processador entre *t=0* e a sua desocupação (fim de qualquer período ocupado). Essa figura indica todos os tempos de invocação e transmissão de cada instância da tarefa 3, podendo-se verificar com mais detalhes na Tabela 7.

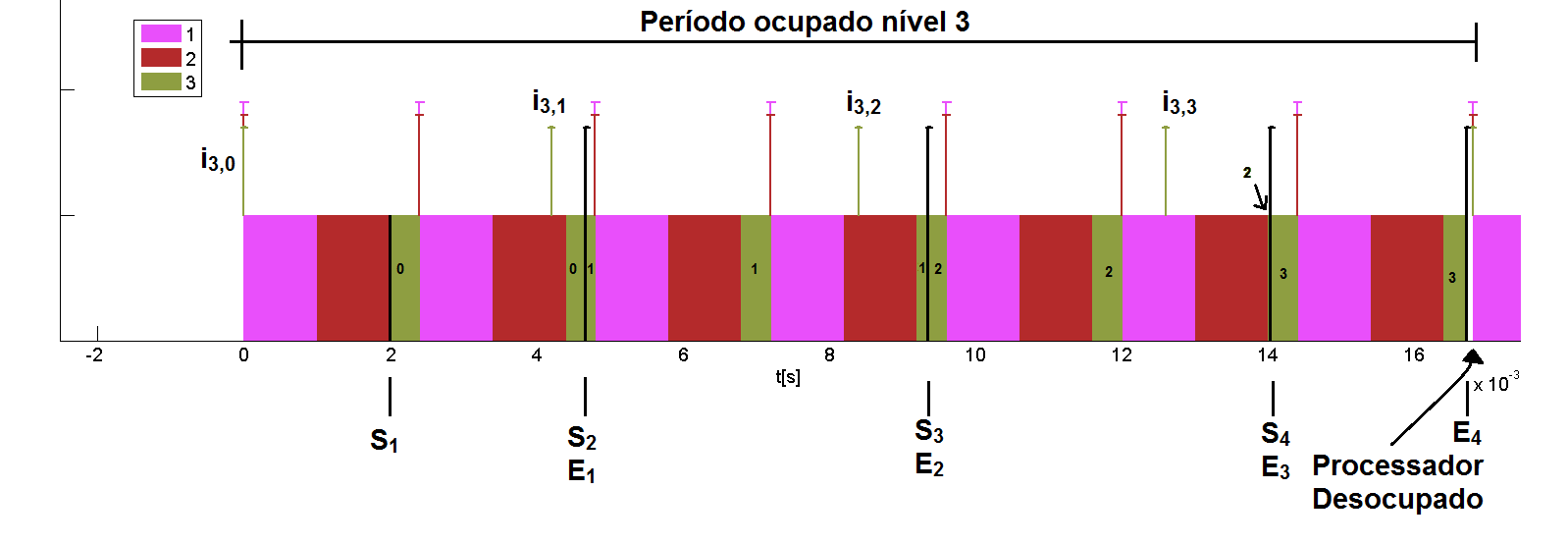


Figura 31 - Período ocupado nível 3 com maior tempo de resposta na segunda invocação

Fonte: Autoria Própria

Tabela 7 – Detalhe dos tempos fornecidos na Figura 31

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instância | Início de ocupação (S) (*ms*) | Fim de ocupação (E) (*ms*) | Tempo de invocação(*ms*) | Tempo de resposta(*ms*) |
| 0 | 2 | 4.68 | 0 | 4.68 |
| 1 | 4.68 | 9.36 | 4.2 | 5.16 |
| 2 | 9.36 | 14 | 8.4 | 5.60 |
| 3 | 14 | 16,7 | 12.6 | 4.10 |

O valor do tempo de resposta da Tabela 6, fornecida pela equação (1) é de 4.68*ms*, o que leva a conclusão é necessário expandir a equação original para os casos em que o tempo de resposta pode ser maior que o período.

### Equação genérica para deadline arbitrário

Para analisar corretamente o tempo de resposta de tarefas com *deadline* arbitrário, deve-se verificar a resposta de cada execução da tarefa *i* dentro do seu período ocupado (cada execução da Tabela 7), expandindo a equação (15) e separando o período ocupado em janelas. Cada uma das janelas tem início no instante (tempo de invocação da instância de *i*) e fim no instante em que a instância acabou de ocupar o processador, como mostrado na Figura 32.

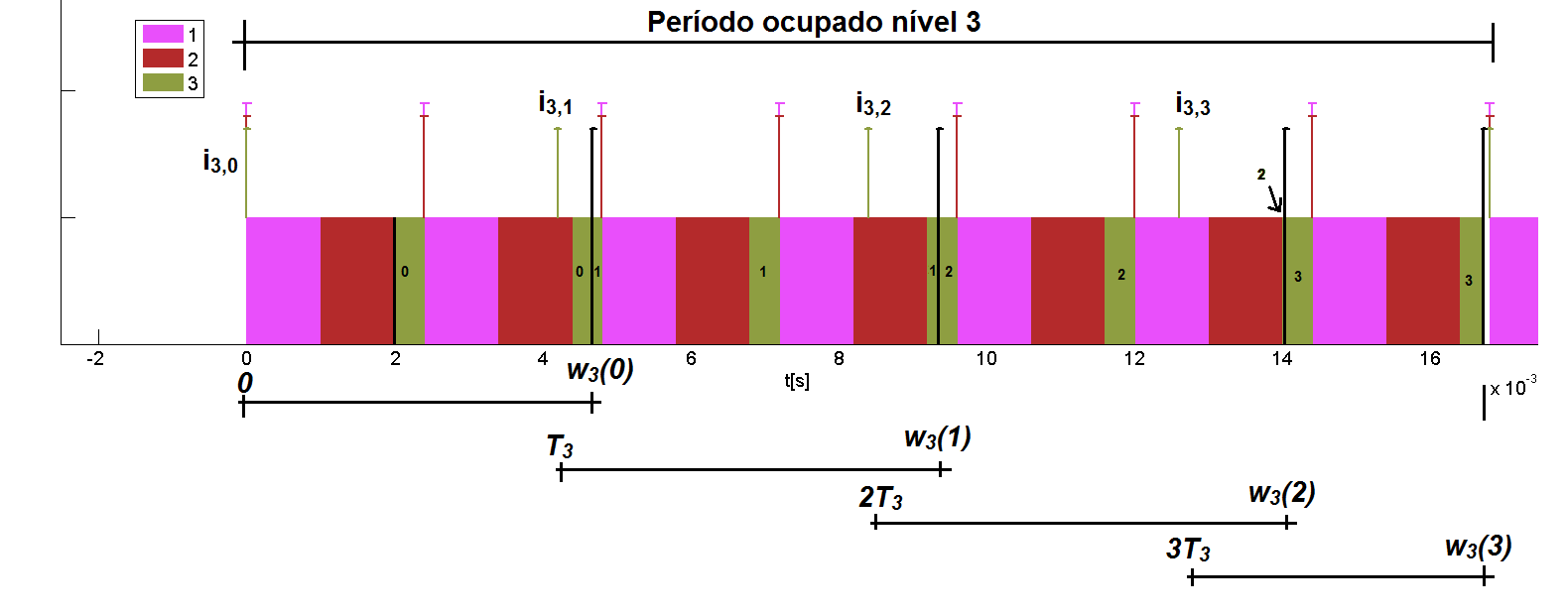


Figura 32 - Período ocupado dividido em janelas de acordo com a tarefa 3

Fonte: Autoria Própria

A equação (22) foi deduzida por Tindell (1992), onde o valor significa o tempo em que *i* ocupou o processador desde o início do período ocupado até o fim da janela , e o somatório constitui no termo referente às interferências causadas por tarefas mais prioritárias durante o mesmo instante, similarmente a equação (15).

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

Similarmente à equação (15), (22) também pode ser avaliada de maneira iterativa, até obter convergência, onde o valor inicial a ser utilizado é apresentado na equação (23).

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

O termo k da equação (17) também teve de ser expandido, pois a análise agora consiste em todas as interferências presentes até o fim da janela definida por *q* e não somente *q*=0 (primeira janela), levando a:

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

O tempo de resposta para cada é igual à diferença entre o fim da execução e o tempo de invocação da mesma instância (início da janela). Para se encontrar o pior tempo de resposta, deve-se achar o maior (pior) tempo de resposta para todos os (eq. (25) e (26)).

|  |  |  |
| --- | --- | --- |
|  |  | (25) |
|  |  | (26) |

Os valores de *q* são números inteiros que são incrementados desde zero até que a relação seja satisfeita.Isto que pode ser compreendido como o momento em que o fim da janela de *q* (fim de execução) é menor que o tempo de sua próxima invocação, não ocorrendo o bloqueio dela mesma na sua próxima instância.

Observa-se que a equação (22), corresponde a forma geral da equação (15) para .

### Tempo de bloqueio

Ao analisar-se um sistema operacional real, Tindell leva em conta que de acordo com o protocolo de prioridade teto (*Priority Ceiling Protocol*), uma tarefa de baixa prioridade pode bloquear uma de mais alta prioridade, alterando o tempo de resposta calculado pelas equações.

Isso significa que em um período ocupado de nível *i*, pode ocorrer o fato de que uma tarefa de prioridade menor que *i* utilize o processador, fazendo-se necessário adicionar um tempo fixo no cálculo de , chamado de tempo de bloqueio (*blocking time*).

O tempo de bloqueio, , corresponde ao tempo de execução da maior seção crítica de um semáforo com prioridade de teto maior ou igual a *i*.

Neste caso, a equação (22) é modificada para:

|  |  |  |
| --- | --- | --- |
|  |  | (27) |

### Jitter de soltura

Tindell inclui na análise de tempo de resposta uma imperfeição prática de alguns sistemas operacionais, o jitter de soltura (*Release Jitter*).

Esse fenômeno consiste em um atraso aleatório entre o tempo em que a tarefa está pronta para ser executada (tempo de chegada) e o instante em que o escalonador é executado (tempo de soltura). Tal fato pode ocasionar um aumento da interferência de tarefas mais prioritárias que uma determinada tarefa *i*, consequentemente aumentando o período ocupado de mesmo nível.

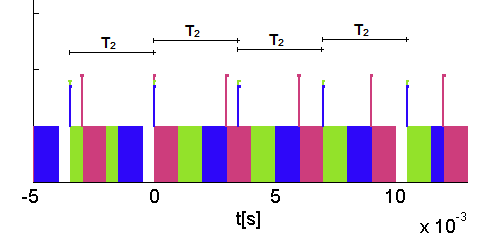


Figura 33 - Exemplo de escalonamento sem efeito do jitter

Fonte: Autoria Própria

A inclusão do efeito do jitter na análise do período ocupado faz com que o momento crítico de nível *i* tenha uma necessidade de ampliação na sua definição, pois, se incluíssemos o efeito do jitter no escalonamento de uma execução da tarefa 2 da Figura 33, obteríamos a Figura 34 como resultado.

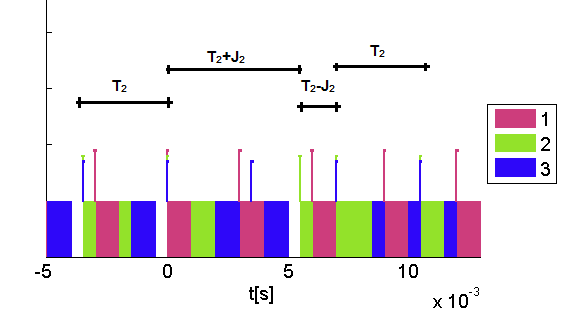


Figura 34 - Exemplo do escalonamento da Figura 33 com efeito do jitter na terceira execução da tarefa 2

Fonte: Autoria Própria

Apesar de as três tarefas terem sido soltas ao mesmo tempo (momento crítico em *t=0*), a Figura 34 não apresenta em *t=0* o início do maior período ocupado de nível 3 (como a Figura 33). Isso ocorre pois o jitter fez com que uma execução da tarefa 2 fosse adiada, diminuindo o número de execuções dessa tarefa até o fim do período ocupado, o que também diminuiu a interferência sofrida pela tarefa 3 durante o período ocupado (entre *t=0* e *t=5ms)*, incluindo-a no período ocupado subsequente.

Para uma análise de pior tempo de resposta da tarefa 3, busca-se o pior período ocupado de nível 3 possível, o qual não foi obtido pela Figura 34, pois nesse cenário houve uma diminuição das interferências de maior prioridade sofridas pela mesma.

O *jitter* considerado consiste em apenas um atraso, o que inicialmente poder-se-ia pensar que existiriam piores tempos de resposta que uma situação em que o mesmo não ocorre, pois por ser um atraso, ele aumenta a distância entre invocações, diminuindo a interferência em uma tarefa *i.* Entretanto, observa-se que há uma “compressão” nos tempos de soltura da Figura 34 entre a terceira e quarta execuções da tarefa 3 (T2-J2).

Essa “compressão” é gerada pelo fato de que ao mesmo tempo em que o *jitter* aumenta a distância entre duas invocações sucessivas (tarefa 3 - invocações 2 e 3 da Figura 34), ao considerar a próxima invocação à qual sofreu *jitter*, houve uma diminuição no tempo entre elas (tarefa 3 - invocações 3 e 4 da Figura 34).

Comparando as figuras Figura 33 e Figura 34, observa-se que ocorreu uma expansão do período ocupado ao qual a terceira execução da tarefa 3 foi transferida. Uma invocação que, sem jitter, pertence ao período ocupado que inicia em *t=0*, ao sofrer o atraso do jitter, pode ser “empurrada” em direção ao próximo período ocupado, se unindo a este, resultando em um caso todavia pior.

O pior caso de interferência de uma tarefa mais prioritária com jitter será quando ela inicie o período ocupado (*t=0*) com uma execução que tenha sofrido efeito do seu pior jitter (maior compressão, portanto maior interferência).

A ocorrência de *jitter* durante o pior período ocupado leva a um caso mais otimista, pois ele distancia execuções sucessivas, podendo resultar em uma “quebra” do período ocupado (fim prematuro do mesmo, como ocorreu na Figura 34).

O *momento crítico de nível i* será quando todas as tarefas mais e igualmente prioritárias a *i* sejam soltas ao mesmo tempo (*t=0*), e essa primeira invocação tenha sido afetada pelo *jitter,* resultando em uma compressão na invocação subsequente. Neste caso, o *jitter* não pode afetar invocações além desta, pois pode dividir o período ocupado.

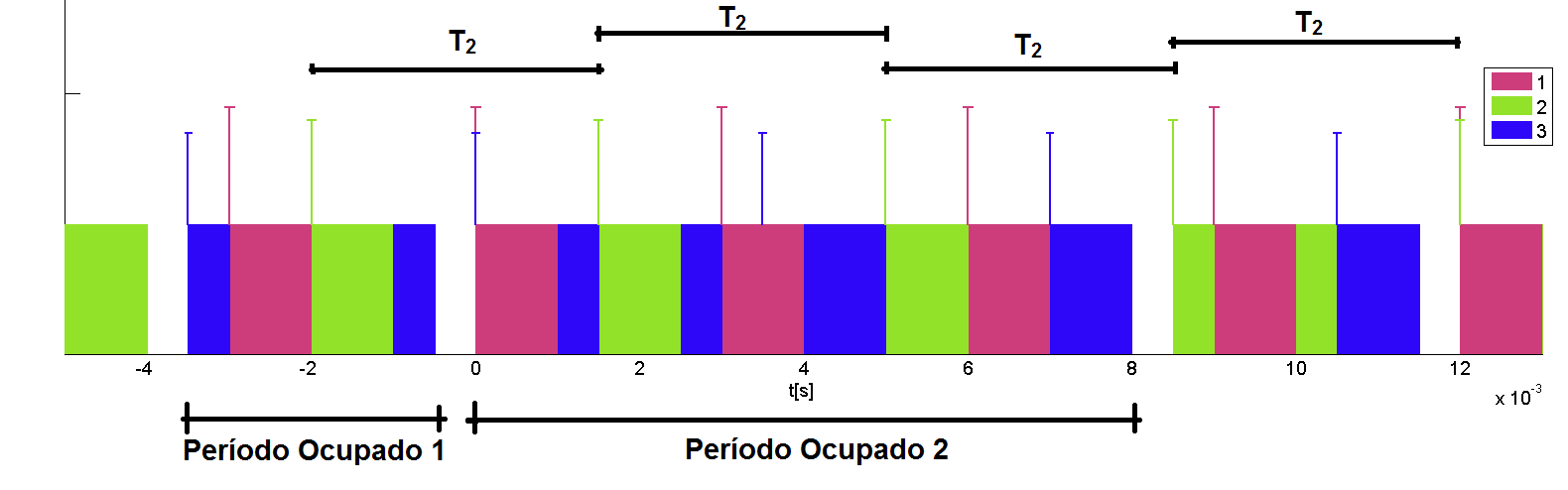


Figura 35 - Sem jitter as execuções do período ocupado 1 não invadem as do periodo ocupado 2

Fonte: Autoria Própria

Acima está ilustrado o mesmo escalonamento da Figura 33, porém, fora de seu momento crítico. Se a execução da tarefa 2 presente no período ocupado 1 sofrer um atraso (jitter) de *2ms*, o resultado será a Figura 36.

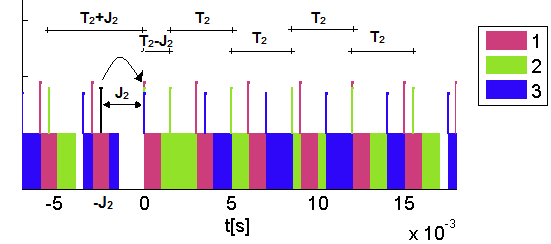


Figura 36 - Início do período ocupado em t=0 por tarefas que sofreram jitter

Fonte: Autoria Própria

A presença de jitter no sistema aumenta o valor do tempo de resposta calculado, pois uma nova invocação de cada tarefa mais prioritária que *i* será incluída no início do pior período ocupado, aumentando a duração de todo o período ocupado.

Devido a possível inclusão de uma nova invocação de tarefas mais prioritárias no início do pior período ocupado, a fórmula da interferência deve ser modificada em função do jitter, *Jj*, de cada tarefa mais prioritária. Agora as invocações presentes durante o período ocupado não correspondem mais a equação (8), pois a tarefa que deveria ter sido solta em *t=-Jj* (Figura 36) sofreu efeito do jitter e foi incluída no período ocupado que começou em *t=0*, levando a um número de invocações *k* fornecido pela equação (28).

A diferença dessa equação (28) para a anterior (24) é que para incluir os efeitos do jitter e avaliar a quantidade de invocações ocorridas até o fim do período ocupado, pode-se interpretar que a duração do período ocupado tenha sido ampliada, como demonstrado na Figura 37.

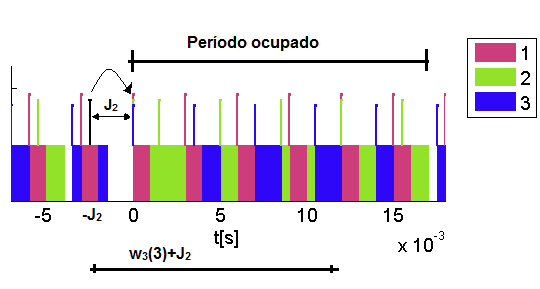


Figura 37 - Indicação do pior período ocupado de nível 3 com jitter na tarefa 2

Fonte: Autoria Própria

|  |  |  |
| --- | --- | --- |
|  |  | (28) |

Essa nova equação faz com que o cálculo do tempo final da janela se torne:

|  |  |  |
| --- | --- | --- |
|  |  | (29) |

Para o cálculo do tempo de atraso final, deve-se somar um possível jitter sofrido pela tarefa que se está analisando, pois para gerar o momento crítico de nível *i* , o tempo de invocação/chegada da tarefa foi em *t=-Ji* , gerando tempos de invocação , levando a equação (30).

|  |  |  |
| --- | --- | --- |
|  |  | (30) |

Similarmente à equação para deadline arbitrário, os valores de *q* são inteiros positivos que são iniciados em zero e vão até o valor anterior ao que satisfizer a equação .

### Tarefas esporadicamente periódicas

Tindell também realizou uma análise de tempo de resposta para tarefas que são esporadicamente periódicas.

As tarefas esporadicamente periódicas consistem em tarefas que são executadas em rajadas, como presente na figura abaixo:

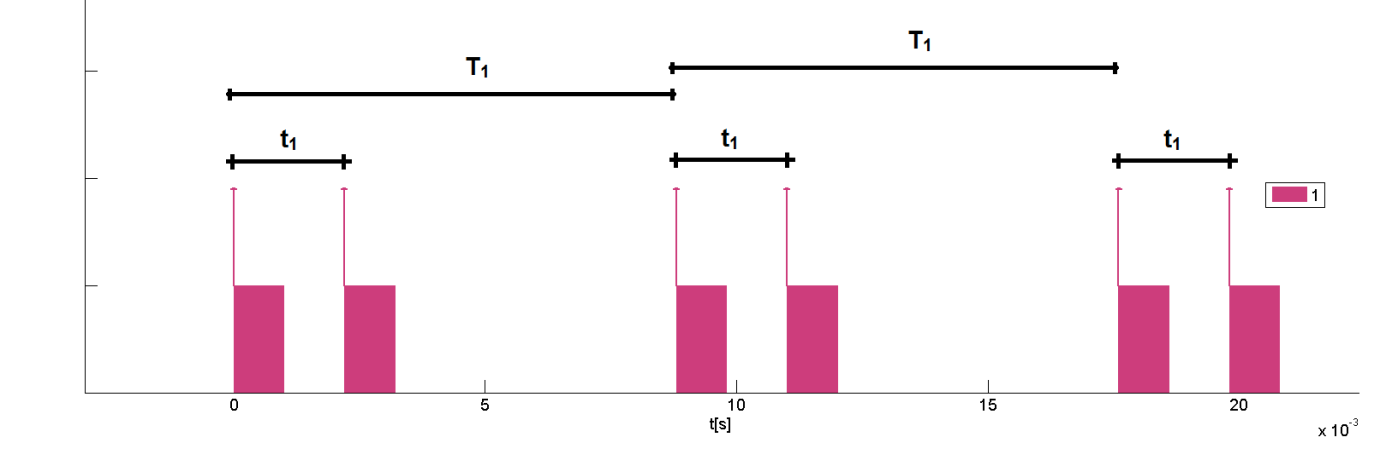


Figura 38 - Tarefa esporadicamente periódica

Fonte: Autoria Própria

No sentido de estratégia de análise, o estudo de escalonabilidade desse tipo de tarefas é muito similar ao anterior, devido à utilização da mesma abordagem de período ocupado.

Tomando como base a Figura 38, serão definidas três variáveis: *Ti* e ti que correspondem, respectivamente, ao período externo e interno de invocação (T1 e *t1*) e *ni* que é a quantidade de invocações de uma tarefa a cada período externo (neste caso *n1=2*).

Caso tenhamos o conjunto abaixo, a Figura 39 nos dá o pior período ocupado que será dividido em janelas para ser analisado.

Tabela 8 - Conjunto de tarefas onde duas são esporadicamente periódicas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *t(ms)* | *T(ms)* | *C (ms)* | *ni* | prioridade |
| 2.0 | 8.0 | 1 | 2 | 1 |
| 2.2 | 17.6 | 1 | 4 | 2 |
| 4.0 | -- | 1 | -- | 3 |

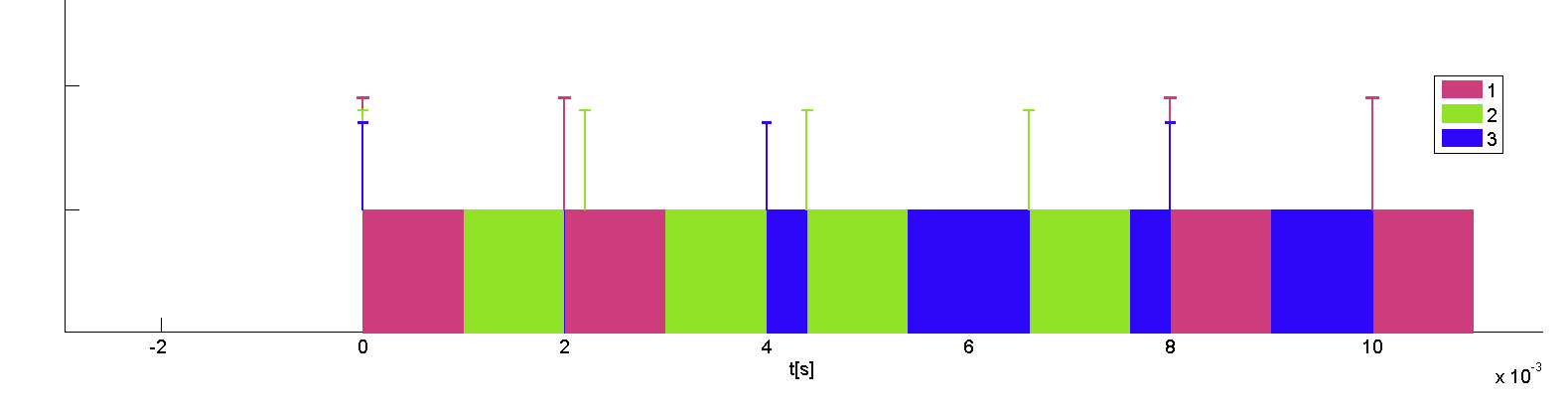


Figura 39 - Pior período ocupado do conjunto dado pela Tabela 8

Fonte: Autoria Própria

Assim como anteriormente, devemos primeiro dividir o pior período ocupado em janelas, como na Figura 40, e analisar a interferência das tarefas de maior prioridade até o fim de cada execução da tarefa analisada.

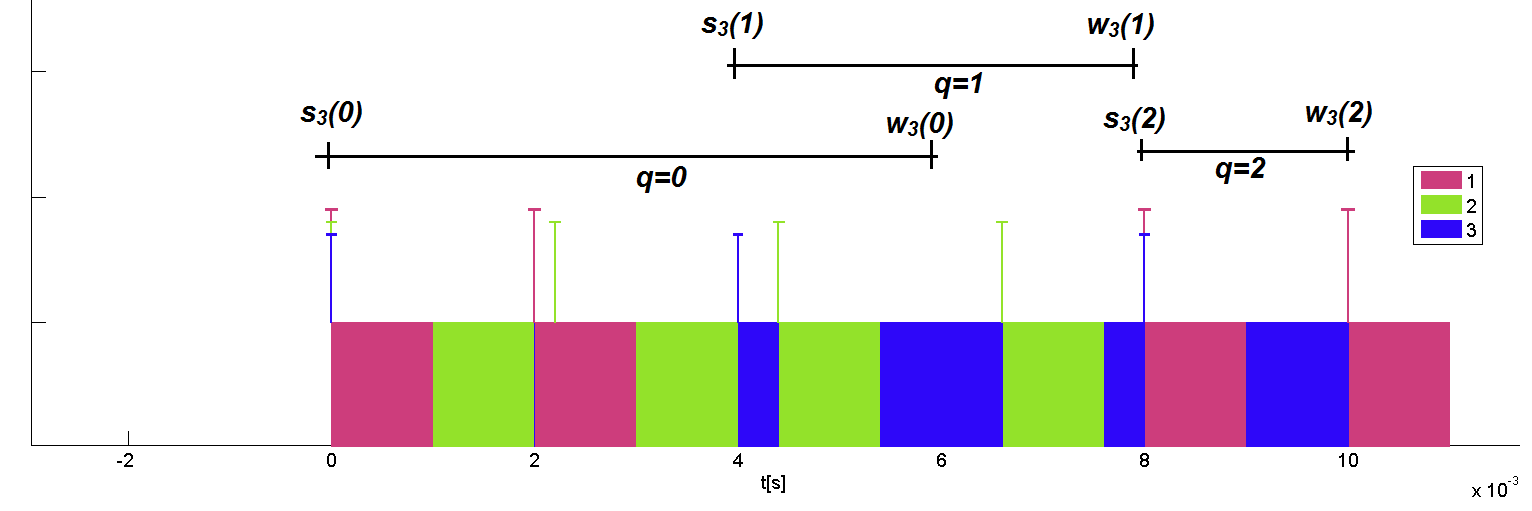


Figura 40 - Divisão em janelas da Figura 39 para análise da tarefa 3

Fonte: Autoria Própria

Na figura acima, temos demarcados o começo, *s3(q)*, e fim, *w3(q)*, de cada janela, onde *q* corresponde à invocação que está sendo examinada.

#### Tempos de invocação

O começo da janela para tarefas periódicas corresponde aos tempos de invocação, *qTi*, de cada instância *q.* Entretanto, sea mensagem for esporadicamente periódica, para encontrar os seus tempos de invocação em função de *q,* devemos modificar um pouco essa relação, pois agora temos dois períodos.

Por exemplo, se quisermos descobrir o tempo da quarta invocação (*q=4*)da tarefa 1 na Figura 40 (*t=10ms)*, devemos analisar utilizar o seu período interno (*2ms*), externo (*8ms*) e o *ni* (2).

Estipula-se que o momento crítico está ocorrendo em *t=0*, assim não há a necessidade de incluir um fator de *offset* de tempo.

Definindo a variável *Mi* como o número de períodos externos que ocorreram antes desse tempo de invocação, temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (31) |

Uma vez que são conhecidos quantos períodos externos ocorreram antes desse tempo de invocação, sabe-se onde começou a invocação da rajada de determinada tarefa. Desta forma pode-se subtrair de *q* a quantidade de invocações que ocorreram até esse início, resultando em uma variável *mi*, que nos fornece o valor de períodos internos que se passaram desde o início da rajada.

|  |  |  |
| --- | --- | --- |
|  |  | (32) |

Através do exemplo fornecido na Figura 40 temos, através das equações (31) e (32), que e , o que pode ser confirmado graficamente pela Figura 41.

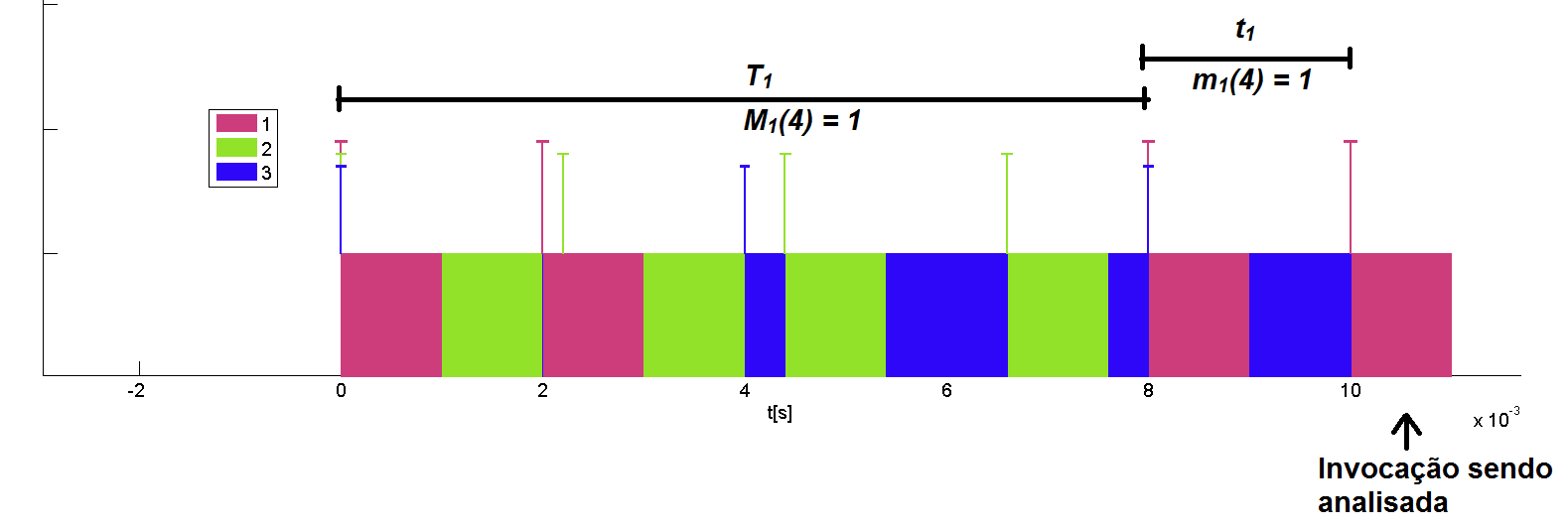


Figura 41 - Estimativa do tempo de invocação de uma mensagem esporadicamente periódica

Fonte: Autoria Própria

Assim, o tempo de invocação, , resulta na seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (33) |

A equação (33) nos fornece o tempo de início da janela a ser analisada. Agora nos resta computar as interferências de mensagens mais prioritárias e da própria mensagem.

#### Interferência de mensagens mais prioritárias

Para equacionar uma interferência esporadicamente periódica devemos encontrar a quantidade de vezes em que essa tarefa mais prioritária, *j*, foi executada desde *t=0* até um tempo *t=wi(q)*, sendo que isso ocorreu durante um período ocupado (cada invocação de *i* resultou em uma execução completa).

O número de períodos externos inteiros de uma tarefa *j* que ocorreram até *wi(q)* é dado por:

|  |  |  |
| --- | --- | --- |
|  |  | (34) |

Tendo-se conhecimento do número de períodos externos inteiros, resta calcular se houve um período externo incompleto, e em caso positivo, determinar a quantidade de períodos internos inteiros que ocorreram até *wi(q)*.

Para isso, subtrai-se *wi(q)* pelo início da última rajada (), o que resultará no intervalo de tempo em que a última rajada esteve executando.

Esse intervalo de tempo terá algumas execuções completas de período interno, portanto, para calcular a quantidade dessas execuções, divide-se pelo período interno, resultando na equação (35).

|  |  |  |
| --- | --- | --- |
|  |  | (35) |

Portanto, somando todas as instâncias executadas de uma tarefa *j* (internas e externas) até *wi(q)*, temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (36) |

Nota-se que a parcela, , correspondente ao número de execuções de período interno na última invocação do período externo, deve ser necessariamente menor que (quantidade de execuções dentro de um período externo), por isso inclui-se a restrição na equação (140), resultando em (37).

|  |  |  |
| --- | --- | --- |
|  |  | (37) |

Tendo-se o número de execuções,, de uma tarefa *j* geradora de interferência, basta-se multiplicar por para obter a quantidade em segundos do atraso provocado.

Assim, generalizando o valor *k* que pertence a equação de interferência (16) e (17), temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (38) |

Finalmente, somando todas as interferências de tarefas mais prioritárias que *i*:

|  |  |  |
| --- | --- | --- |
|  |  | (39) |

#### Interferência da mesma mensagem

O número de períodos externos e internos que ocorreram antes da tarefa *i* são dados, respectivamente, pelas equações (31) e (32), através das quais se pode calcular a quantidade de vezes,, em que a tarefa *i* ocupou o processador anteriormente ao início de sua janela (dado pela equação (33)).

|  |  |  |
| --- | --- | --- |
|  |  | (40) |

Para mensurar toda a interferência causada pelas suas execuções, deve-se incluir uma unidade à equação (40), pois ela somente corresponde às execuções anteriores ao início da janela.

A variável corresponde a quantidade de execuções, por isso, há a necessidade de multiplicar pelo tempo de cada execução, , levando a equação (41).

|  |  |  |
| --- | --- | --- |
|  |  | (41) |

#### Fim da janela

As análises feitas nos tópicos anteriores levaram às equações (39) e (41), que, em conjunto, fornecem os atrasos causados por execuções da própria mensagem e de mensagens mais prioritárias, resultando na seguinte equação para o final da janela de uma tarefa *i* e instância:

|  |  |  |
| --- | --- | --- |
|  |  | (42) |

A equação acima não contempla o atraso provocado pelo tempo de bloqueio gerado por tarefas menos prioritárias descrito na seção 6.1.5, assim, a equação (42) torna-se:

|  |  |  |
| --- | --- | --- |
|  |  | (43) |

#### Tempo de resposta

Para obter o tempo de resposta a partir da equação (43) deve-se obter novamente a Figura 40, onde está indicada a relação entre o fim da janela e o tempo de invocação da instância *q* da tarefa analisada (começo da janela)*.*

Portanto, o tempo de resposta final resulta da subtração da equação (43) pelo tempo de invocação da instância *q*, dado por (33).

|  |  |  |
| --- | --- | --- |
|  |  | (44) |

O procedimento para calcular o pior tempo de resposta é muito semelhante ao anterior (fim da seção 6.1.4). Deve-se calcular vários tempos de resposta com *q=0,1,2,3,...* até o valor anterior ao que satisfizer a relação .

#### Jitter

Na seção anterior, a análise do tempo de resposta foi generalizada para situações em que existem tarefas esporadicamente periódicas, porém, que não sofrem o efeito do jitter descrito em 6.1.6.

Através de um procedimento muito similar ao feito para tarefas periódicas (6.1.6), também podemos expandir as equações (43) e (140), obtendo o tempo de resposta de uma tarefa esporadicamente periódica com jitter.

A geração do *momento crítico de nível i* não sofreu alteração para a análise desse novo tipo de mensagens, o que significa que para que o efeito do jitter seja considerado, basta fazer com que a primeira instância das tarefas sofra o pior jitter possível e que elas sejam liberadas ao mesmo tempo, igualmente ao procedimento foi realizado na seção 6.1.6. Portanto, a equação (34), que determinava quantos períodos externos ocorreram até um determinado momento , deve incluir o jitter através de um aumento da janela que se está sendo analisada, pois uma instância do período ocupado anterior pode fazer parte do pior período ocupado.

|  |  |  |
| --- | --- | --- |
|  |  | (45) |

Consequentemente, a equação (35) também deve ser alterada:

|  |  |  |
| --- | --- | --- |
|  |  | (46) |

A equação do tempo de resposta final (140) também deve ser alterada, como feito em (30), pois a tarefa foi solta segundos após o seu momento de chegada.

|  |  |  |
| --- | --- | --- |
|  |  | (47) |

Para encontrar os valores possíveis de *q*, a mesma relação se mantém (), porém, o tempo de invocação para as mensagens com jitter não obedece à equação (33), pois para gerar o momento crítico de nível *i* supõe-se que a primeira mensagem chegou em *t=-Jj*, o que adiciona um *offset* a cada invocação:

|  |  |  |
| --- | --- | --- |
|  |  | (48) |

### Teste completo de escalonabilidade

O teste completo de escalonabilidade de uma tarefa esporadicamente periódica *i* com jitter é dado através da solução das equações deduzidas anteriormente e reapresentadas abaixo.

Essa análise consiste em um processo iterativo, onde itera-se a variável inteira e positiva *q* (*q = 0,1,2,3,...)* enquanto a seguinte condição se satisfizer:

|  |  |  |
| --- | --- | --- |
|  |  | (49) |

No início de cada iteração de *q* (iniciada com *q=0)* calculam-se as variáveis e das equações (50) e (51), as quais serão utilizadas como parâmetros nas equações seguintes. Posteriormente, através das equações (52) calcula-se o valores de para cada tarefa *j* mais prioritária que *i*, o qual será utilizado para o cálculo de , onde o índice *n* também representa mais um laço de iteração.

|  |  |  |
| --- | --- | --- |
|  |  | (50) |

|  |  |  |
| --- | --- | --- |
|  |  | (51) |

|  |  |  |
| --- | --- | --- |
|  |  | (52) |

A iteração de *n* é iniciada em zero e terminada na convergência de, ou seja, quando .

Nas equações apresentadas, as variáveis, , e são parâmetros fixos de uma tarefa *k*, os quais consistem, respectivamente, em período externo, período interno, tempo de bloqueio, número de instância dentro do período externo e tempo de execução.

|  |  |  |
| --- | --- | --- |
|  |  | (53) |

|  |  |  |
| --- | --- | --- |
|  |  | (54) |

Após a convergência do valor de , calcula-se o tempo de resposta de uma determinada instância *q* da tarefa *i*, , através da equação (54). Esse valor não será o pior tempo de resposta, sendo que para que este seja obtido, cada calculado através da iteração de *q* deve ser comparado, e o maior valor destes consistirá no pior tempo de resposta, .

|  |  |  |
| --- | --- | --- |
|  |  | (55) |

Portanto, a análise de Tindell nos fornece um método iterativo que calcula o pior tempo de resposta possível para tarefas esporadicamente periódicas ou integralmente periódicas, com ou sem a presença jitter.

## ADAPTAÇÃO DAS EQUAÇÕES DE ESCALONAMENTO AO BARRAMENTO CAN

A política dinâmica de escalonamento do barramento CAN é de um sistema de prioridade fixa não preemptivo, enquanto vários sistemas operacionais utilizam um escalonamento de prioridade fixa, porém, preemptivo, que teve a sua análise de tempo de resposta avaliada por Tindell (1992) e está descrita na seção anterior.

Devido à semelhança entre essas políticas de escalonamento, Tindell e Burns (1994) adaptaram as equações desenvolvidas (para deadline menor que período) ao barramento CAN. Com isto, forneceram um método de cálculo de pior tempo de resposta para um conjunto de mensagens periódicas com jitter, avaliando também o impacto do comportamento não ideal do controlador CAN Intel 82527.

O jitter considerado por Tindell e Burns (*queuing jitter*) deve-se à variabilidade no tempo de enfileiramento de uma mensagem dentro da própria ECU, que é o tempo para que a mesma seja enviada ao hardware do controlador e participe da próxima arbitragem de barramento. Este fenômeno ocorre em todas as mensagens, sendo, porém, mais perceptível em mensagens periódicas, uma vez que pode-se estimar as suas invocações somente através da observação de suas transmissões no barramento.

A variabilidade ocorre devido ao fato de que o software das ECUs geralmente são implementados com a utilização de sistemas operacionais, devido à necessidade de atendimento de várias atividades. Dependendo de que tarefas estão sendo executadas, um ou outro atraso pode ocorrer nesse enfileiramento.

Tindell e Burns também realizam a análise de pior tempo de resposta (*benchmark*) de um conjunto que inclui mensagens periódicas e não periódicas, assumindo valores hipotéticos de jitter a todas as mensagens e hipotéticos de período às mensagens que não são periódicas, gerando implicações que serão vistas em seções subsequentes.

Davis, Burns, Bril e Lukkien (2007) revisaram a teoria de escalonamento aplicado a CAN e identificaram um problema na análise proposta por Tindell e Burns(1994), corrigindo-a através de um teste de escalonabilidade um pouco mais complexo.

### Diferenças entre preemptivo e não preemptivo

Para iniciar a relação feita por Tindell e Burns (1994) entre a análise de escalonamento de sistemas operacionais e o barramento CAN, devemos observar se a equação previamente deduzida pode ser aplicada à um sistema não preemptivo.

A diferença entre um escalonador preemptivo e um não preemptivo é de que o preemptivo interrompe a execução da tarefa atual se uma tarefa de prioridade mais alta estiver na fila de execução. Ou seja, seria o caso em que uma tarefa de baixa prioridade fosse interrompida durante a sua execução para que uma de mais alta prioridade pudesse ser executada, e após isso, a execução da tarefa de baixa prioridade fosse retomada do ponto em que havia parado.

Um escalonador não preemptivo não permite a interrupção de uma tarefa que já está sendo executada, independente de sua prioridade, existindo um momento específico para a execução de uma nova mensagem.

No barramento CAN, o momento de arbitragem marca a decisão de qual mensagem vai ser transmitida em função de sua prioridade. Uma vez que uma mensagem tenha sido arbitrada, ela é transmitida por completo independentemente de sua prioridade.

A equação de interferência utilizada para o caso em que o deadline é menor que o período (16) permanece quase a mesma, uma vez que o fato de existir ou não a preempção somente altera a ordem em que as tarefas de mais alta prioridade serão executadas. Não se altera o fato de que elas vão necessariamente ser executadas antes do início da mensagem que está sendo analisada.

Entretanto, para computar a quantidade de vezes que uma tarefa de prioridade mais alta interrompe uma de menor prioridade, a equação (17) será levemente modificada para a (56).

|  |  |  |
| --- | --- | --- |
|  |  | (56) |

Essa alteração foi feita pois por mais que as equações de interferência sejam muito parecidas, a variável foi substituída por , já que não estamos mais buscando as interferências que ocorreram até o momento de fim de execução, mas as que ocorrem até o inicio, já que qualquer invocação depois do início de execução deve ser desconsiderado, conforme na figura abaixo:

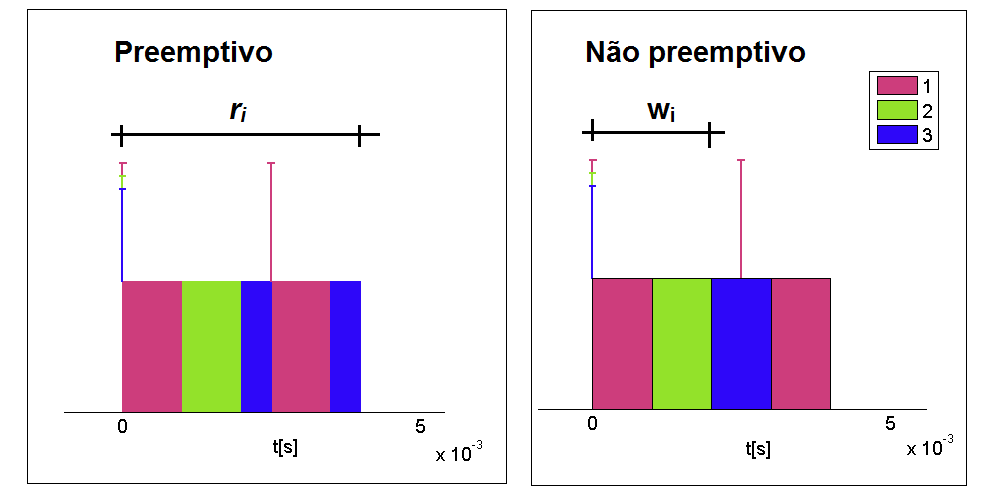


Figura 42 - Diferença entre um momento crítico preemptivo e outro não preemptivo

Fonte: Autoria Própria

Em um sistema preemptivo buscamos o fim de execução da tarefa, pois o seu momento de início de execução não tem um significado relevante para o cálculo do tempo de resposta e as invocações de tarefas prioritárias durante a execução devem ser consideradas, pois todas corresponderão a uma interferência. Já num sistema não preemptivo, sabendo-se o início de execução, tem-se facilmente seu fim e ignora-se qualquer invocação que possa ocorrer depois desse inicio.

Essa modificação causará alguns leves impactos nas equações já deduzidas anteriormente, como por exemplo, o tempo de resposta final agora é dado pela equação abaixo:

|  |  |  |
| --- | --- | --- |
|  |  | (57) |

Onde, assim como antes, corresponde ao pior jitter sofrido pela mensagem, , ao seu tempo de transmissão e ao início de transmissão.

### Tempo de transmissão

Previamente consideramos que o tempo de execução ,, de cada tarefa, era fixo, porém, quando transpomos essa análise às mensagens do barramento CAN, verificamos que devido aos *bit-stuffing* (seção 2.4.4) o tamanho das mensagens varia de acordo com os dados que nela são transportados.

Uma vez que estamos tratando de pior caso de tempo de resposta, é razoável utilizar o pior caso de tempo de transmissão, ou seja, calcular o pior possível e utilizá-lo durante toda a análise.

Tindell e Burns (1994) fornecem a seguinte relação para encontrar o pior valor de em função da quantidade de bytes de dados, , a serem enviados na mensagem e do tempo de bit, :

|  |  |  |
| --- | --- | --- |
|  |  | (58) |

### Interferência de mensagens mais prioritárias

O momento em que corresponde ao início de transmissão de uma mensagem *i*, a qual sofreu interferências de mensagens mais prioritárias até , o que, como deduzido anteriormente (18), resulta na equação :

|  |  |  |
| --- | --- | --- |
|  |  | (59) |

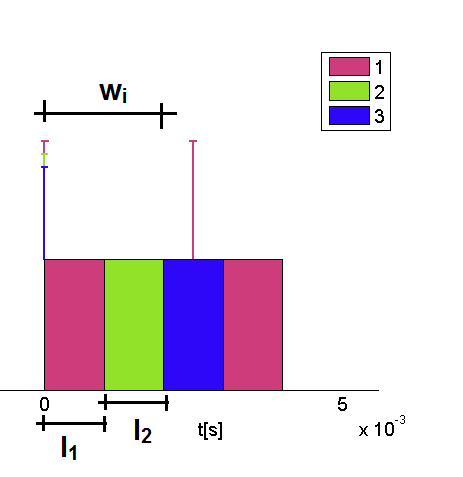


Figura 43 - Soma das interferências até wi (I1+I2) resultam em wi

Fonte: Autoria Própria

Como estamos considerando o momento crítico (todas transmissões em *t=0*) e até o instante somente mensagens mais prioritárias foram transmitidas continuamente, obtemos que a soma das interferências até o início de transmissão resulta no tempo de início de transmissão (), semelhantemente ao que foi feito em (19) e ilustrado na Figura 43.

A equação abaixo é deduzida a partir da condição , fornecendo o tempo de início de transmissão para mensagens com o deadline menor que o período, considerando somente interferências de mais prioritárias.

|  |  |  |
| --- | --- | --- |
|  |  | (60) |

### Tempo de bloqueio

Assim como para as tarefas no sistema preemptivo temos um tempo de bloqueio gerado a partir de tarefas menos prioritárias (6.1.5), esse conceito também existe para mensagens no barramento CAN.

O momento crítico ainda resulta no momento em que existe maior interferência de mensagens de mais alta prioridade. No entanto, se o momento crítico ocorrer em *t=0* e uma mensagem de menor prioridade estiver sendo transmitida, ela será completamente transmitida, devido a não preemptabilidade do barramento, o que gera um atraso proveniente de uma mensagem de menor prioridade, não contemplado pela equação da interferência.

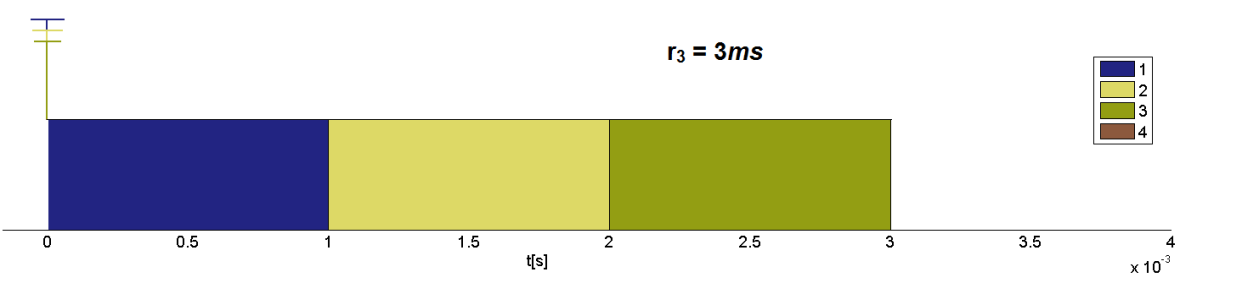


Figura 44 - Mensagens de maior prioridade (1,2 e 3) sem o tempo de bloqueio

Fonte: Autoria Própria

O pior caso de bloqueio é quando mensagem bloqueadora (baixa prioridade) é invocada um pouco antes das invocações de momento crítico, ganhando a arbitragem e bloqueando a transmissão de mensagens com prioridade maior que a dela, como mostrado na Figura 45.

Observa-se que somente uma instância de mensagem de prioridade mais baixa pode realizar esse tipo de bloqueio, pois ela só é transmitida devido ao fato de que as invocações do momento crítico ocorrem durante a sua transmissão, e após o seu fim uma mensagem mais prioritária necessariamente ocupará o barramento.

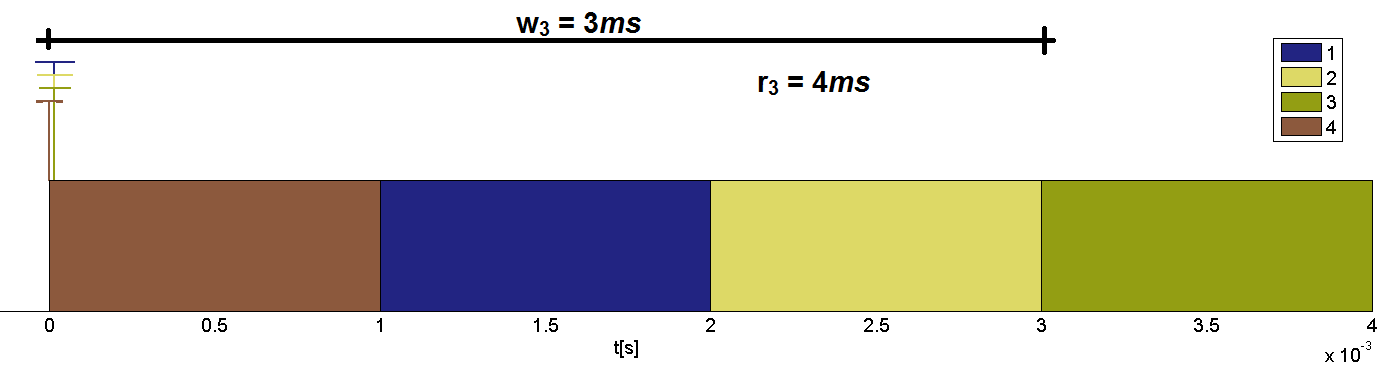


Figura 45 - Mensagens de maior prioridade (1,2 e 3) foram bloqueadas pela mensagem de prioridade mais baixa (4)

Fonte: Autoria Própria

Para adicionar esse tipo de efeito na equação do início de transmissão, devemos somar esse bloqueio,, ao restante das interferências, levando à seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (61) |

A variável caracteriza o bloqueio devido a uma instância de mensagem menos prioritária que *i*, significando que, para encontrarmos o pior caso de tempo de resposta precisamos encontrar o maior valor possível de bloqueio. Portanto, como somente pode ser gerado por mensagens menos prioritárias que *i*, temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (62) |

Em (62), corresponde ao conjunto de mensagens menos prioritárias que *i* (*lower priority – lp*), e é o tempo de transmissão da mensagem.

### Jitter e tempo de bit

Como citado anteriormente, o jitter, ou *queuing jitter*, deve ser incluído na análise de pior tempo de resposta, similarmente ao que foi feito nas seções 6.1.6 e 6.1.7.6, onde a compressão causada pelo jitter fez com que a interferência de tarefas mais prioritárias aumentasse.

O tempo de bit (resolução do barramento) também deve ser considerado. Explica-se: devido à sincronização do barramento, um atraso de mesmo valor pode ser inserido nas transmissões.

Assim, obtemos as seguintes equações, fornecidas por Tindell e Burns (1994), para o cálculo do pior tempo de resposta de uma mensagem CAN:

|  |  |  |
| --- | --- | --- |
|  |  | (63) |

|  |  |  |
| --- | --- | --- |
|  |  | (64) |

|  |  |  |
| --- | --- | --- |
|  |  | (65) |

A resolução da equação (63) é feita através de um método iterativo, idêntico ao que foi utilizado para a (53), e com .

Antes da existência dessa abordagem as taxas de utilização dos barramentos eram de até 40%, passando por diversos testes de confiabilidade para certificar o seu funcionamento. Através do trabalho de Tindell as taxas de utilização puderam ser elevadas até 80%, e foram motivados diversos outros trabalhos. (Davis, et al., 2007).

### Falha na análise de tempo de resposta

Em 2007, Davis, Burns, Bril e Lukkien, apresentaram um trabalho (Davis, et al., 2007) no qual mostraram uma falha no teste de escalonabilidade de Tindell e a corrigiram através de uma extensão na abordagem inicial.

Para evidenciar a falha, utilizaram o seguinte conjunto de mensagens (Tabela 9):

Tabela 9 - Conjunto de mensagens que tem falha na análise do tempo de resposta

|  |  |  |
| --- | --- | --- |
| Prioridade | T | C |
| 1 | 2.5ms | 1ms |
| 2 | 3.5ms | 1ms |
| 3 | 3.5ms | 1ms |

Através das equações de Tindell (63), (64) e (65), o resultado do pior tempo de resposta para a mensagem 3 é de 3ms, porém, como se pode observar na Figura 46, o pior tempo de resposta é de 3.5ms, o que mostra que a análise tem uma falha.

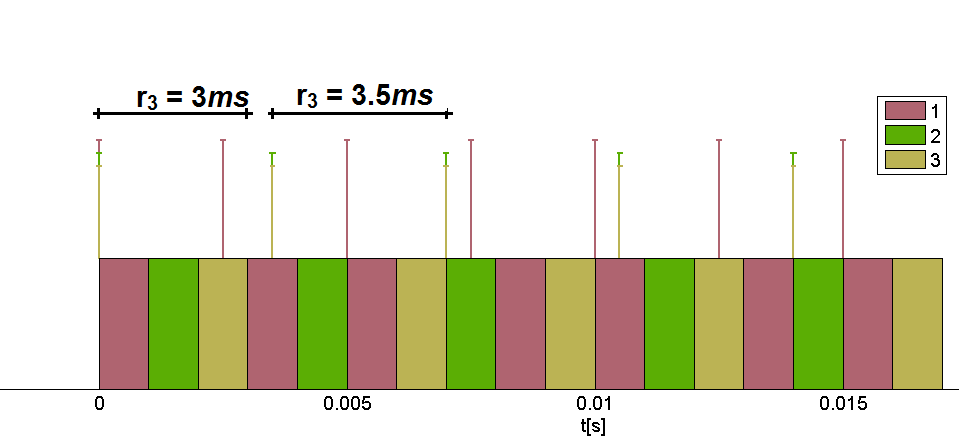


Figura 46 – Período ocupado de nível 3 gerado pelo momento crítico 3 da Tabela 9

Fonte: Autoria Própria

A análise de Tindell para uma mensagem *i* nos dá o tempo de resposta da *primeira* instância transmitida durante o período ocupado. Para o sistema preemptivo e com tarefas que tinham o tempo de resposta menor que o período, era válido considerar que a primeira instância de uma tarefa durante o período ocupado constituía como o pior tempo de resposta.

Essa mesma hipótese foi considerada válida para o sistema não preemptivo, o que é falso, mesmo que o tempo de resposta da mensagem seja menor que o seu período.

No sistema não preemptivo as mensagens que são invocadas durante a transmissão da mensagem *i* devem esperar pela sua arbitragem. Tal fato possibilita a continuação do período ocupado após a primeira instância de *i* e a inserção de bloqueio de uma instância de *i* em sua próxima. Assim faz-se necessária a análise de todas as suas instâncias durante o seu período ocupado.

Observa-se que esse problema também existiu para sistemas preemptivos, porém, quando o tempo de resposta era maior que o período, pois uma instância da tarefa *i* introduzia um bloqueio na execução de uma próxima invocação da mesma tarefa.

### Correção na análise de tempo de resposta

Para corrigir a análise anterior, todas as instâncias dentro do período ocupado precisam ser analisadas, e para isso, uma modificação da definição de período ocupado foi necessária.

#### Período Ocupado

O conceito de período ocupado, introduzido por Lehoczky (1990) e descrito na seção 6.1.3 , foi modificado por Harbour, Klein e Lehoczky (1991), para ser utilizado em sistemas preemptivos com prioridades fixas, porém, que variam durante a execução.

Davis et al. (2007) adaptaram essa modificação ao barramento CAN, definindo o período ocupado de nível *i* como:

1. Tem o seu início em ts quando uma mensagem de prioridade maior ou igual a *i* é invocada, e não existem outras mensagem com prioridade maior ou igual a *i* esperando pela sua transmissão invocadas estritamente antes de ts.
2. É um intervalo contíguo de tempo em que mensagens com prioridade menor que *i* não são transmitidas.
3. Acaba no tempo te ,quando o barramento fica pronto para uma nova arbitragem, porém, nenhuma mensagem de prioridade *i*  ou maior está esperando por transmissão.

Do ponto de vista matemático, o período ocupado pode ser entendido como o intervalo [ts, te) de invocação de mensagens mais ou igualmente prioritárias a *i*.

Essa nova definição de período ocupado é muito parecida com a de Lehoczky. Esta acabava, porém, por unificar dois períodos ocupados (nova definição), resultando em uma análise de mais instâncias que o estritamente necessário.

#### Fim do período ocupado

Sabendo previamente o fim do período ocupado, pode-se descobrir quantas vezes uma determinada mensagem foi transmitida e calcular o tempo de resposta sem a necessidade de checar uma condição , como feito no fim da seção 6.1.4.

Partindo de deduções parecidas com as anteriores, a seguinte equação fornece o tempo de fim do período ocupado de nível *m*, , com .

O conjunto corresponde a todas as mensagens com prioridade maior ou igual a *m*. A equação abaixo fornece o primeiro instante de transmissão de uma mensagem menos prioritária que *m*, que é onde acaba o período ocupado.

|  |  |  |
| --- | --- | --- |
|  |  | (66) |

Para encontrar a quantidade de instâncias transmitidas basta aplicar a seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (67) |

#### Tempo de resposta de cada janela

Similarmente ao que foi feito na seção 6.1.4, dividimos o período ocupado em janelas, porém, agora o valor de interesse não é o fim da janela, mas o seu começo (seção 6.2.1).

Assim, expandindo a equação (63) para cada instância *q* (primeira instância corresponde a *q=0*), temos que:

|  |  |  |
| --- | --- | --- |
|  |  | (68) |

Sendo o tempo de resposta:

|  |  |  |
| --- | --- | --- |
|  |  | (69) |

E o pior tempo de resposta é dado pelo maior valor de tempo de resposta de *q=0* até *q=Qm-1*.

|  |  |  |
| --- | --- | --- |
|  |  | (70) |

## CARACTERÍSTICAS DA ANÁLISE DE PIOR TEMPO DE RESPOSTA

A análise do pior tempo de resposta explorado nas seções 6.1 e 6.2 tem importantes características que podem motivar tanto o seu uso como a busca de outra alternativa.

O resultado fornecido por essa análise é o pior tempo de resposta possível de um determinado conjunto de mensagens, utilizando sempre o pior caso de todas os fatores influentes no tempo de resposta, como utilizar sempre o maior tamanho possível de mensagem e o pior valor de jitter para uma mensagem periódica.

Ao considerar o pior tempo de resposta como base para um limiar de tempo máximo, estamos procurando dimensionar o sistema sempre pelo pior caso, mesmo que isto ocorra uma vez por ano, levando a um superdimensionamento do sistema devido a uma abordagem muito pessimista de análise.

O tempo de resposta fornecido pela equação torna-se, todavia, mais pessimista quando incluímos as mensagens não periódicas nas equações de escalonamento. Estas, se tratadas sempre com o seu pior caso de período (menor intervalo possível entre duas sucessivas), vão tornando a analise ainda vez mais pessimista, superdimensionando o sistema e falhando em fornecer uma avaliação confiável do comportamento do mesmo.

Pode-se considerar que redes CAN tem uma certa robustez à quebra de *deadline* por algumas mensagens, pois mesmo que este fato ocorra, o impacto no sistema pode ser insignificante, sendo mais importante considerar quantas vezes uma mensagem pode quebrar o seu *deadline*. Evidentemente, isso varia de mensagem para mensagem, e algumas mensagens, como as provenientes da ECU do motor, não podem extrapolar os seus *deadlines* em hipótese alguma.

Considerando esses fatos, existe uma motivação ainda maior na busca de uma abordagem que não forneça o pior caso, mas um resultado que forneça uma relação entre a probabilidade de ocorrência dos tempos de resposta, levando a uma probabilidade de quebra de deadline.

A Figura 47 apresenta um gráfico com a relação entre os valores fornecidos pela equação, pior caso de simulação e o tempo de resposta de 98% das mensagens.

Esse gráfico foi obtido através do uso de um simulador programado em MATLAB, o qual demonstra a baixa probabilidade de pior caso, e o fato de 98% das mensagens atenderem ao deadline (*10ms*). Analisando o gráfico, observa-se que ambas as análises de pior caso ultrapassam o deadline. Isto, porém, não acontece com 98% das mensagens, um valor de grande utilidade ao arquiteto do sistema, que através da importância da mensagem e do impacto que a sua quebra de *deadline* causaria, pode julgar se é uma probabilidade aceitável ou não.

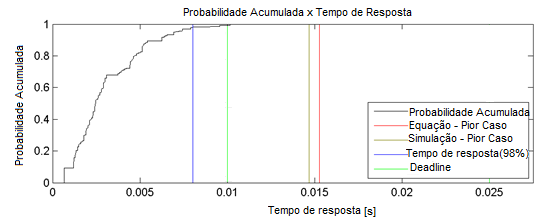


Figura 47 - Gráfico representando a relação de probabilidade entre as diferentes análises

Fonte: Autoria Própria

A robustez à perda de deadline de cada mensagem CAN é diferente, pois envolve uma relação entre quais módulos estão se comunicando e a função da mensagem no sistema como um todo. Uma abordagem probabilística, portanto, é mais interessante pois provê uma forma mais específica de analisar cada mensagem individualmente.

## ANÁLISE ESTATÍSTICA DO BARRAMENTO CAN

O barramento CAN real apresenta muitas imperfeições difíceis de serem analisadas e modeladas isoladamente. As equações apresentadas por Davis et al. só podem ser usadas para cenários muito bem controlados, onde os parâmetros da rede e as informações das mensagens devem ser conhecidos previamente.

Para uma análise mais realista do barramento, levando em consideração as suas imperfeições é proposto neste trabalho um método de análise através de características estatísticas.

O método de análise estatística do barramento CAN não exige conhecimento prévio de suas características, mas é necessário o acesso a um *log* de dados, que contenha as marcações de tempo de transmissão de cada mensagem, o ID das mensagens e o tamanho de cada mensagem.

Como a análise estatística exige um *log* de dados, o barramento deve estar funcionando para que eles sejam coletados. Sendo assim, este método não pode ser usado em fases iniciais de projeto pois não existe um meio para a extração de dados.

Uma forma de coletar os dados necessários é através de dispositivos que são colocados no barramento e o “escutam” sem interferir diretamente, estes dispositivos guardam as informações em memórias não voláteis para análise posterior. Estes dispositivos são chamados de “*Loggers*” e podem ser encontrados em diversas marcas e formatos.

O método proposto tem como objetivo obter uma curva de probabilidade acumulada do tempo de resposta de cada mensagem, para que a probabilidade de uma mensagem passar do deadline possa ser analisada.

Idealmente os tempos de resposta podem ser obtidos projetando os períodos das mensagens para frente e subtraindo o tempo final de transmissão do tempo projetado. Entretanto, o barramento real apresenta nível de *jitter* alto, mensagens aperiódicas e imperfeições no funcionamento dos controladores, como inversões de prioridade nas filas de envio, fazendo com que a forma ideal não possa ser usada.

Para obter o provável tempo de resposta das mensagens é preciso obter os seus tempos de invocação e para isso existe a necessidade de se introduzir o conceito de tempo ocupado.

### Tempo ocupado

Uma mensagem *k* que compete pela utilização do barramento CAN pode ser atrasada por mensagens mais prioritárias ou por uma menos prioritária que já estava sendo transmitida no momento de invocação da mensagem *k*.

O tempo ocupado da mensagem *k* em uma instância *j* qualquer () é dado pelo tempo em que as mensagens mais prioritárias (), ou uma mensagem menos prioritária () tenha ocupado o barramento antes da transmissão da mensagem *k*, ou seja, é o tempo em que a mensagem poderia ter entrado, mas foi atrasada por outras mensagens.

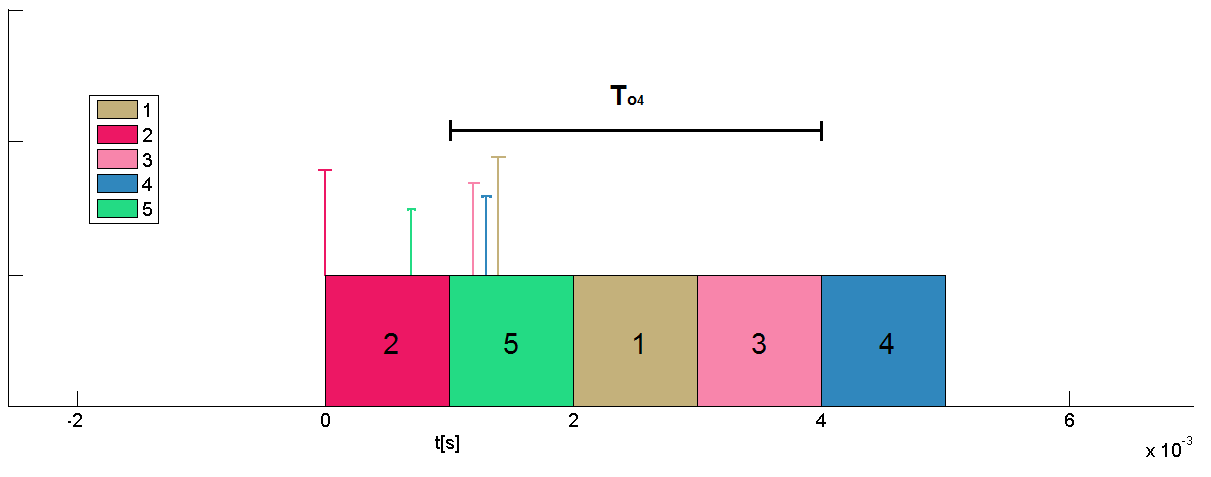


Figura 48 – Tempo ocupado mensagem 4.

Fonte: Autoria Própria

No exemplo da Figura 48, a mensagem de ID 4 foi bloqueada por mensagens de maior prioridade (3 e 1) e por uma de menor prioridade (5). Se a mensagem 4 tivesse sido invocada no tempo de transmissão da 2 então ela teria sido transmitida antes da 5, por isso a 2 não está no período ocupado da 4 nesse exemplo.

As invocações que apresentam o tempo ocupado devem ser descartadas na hora de estimar o período das mensagens, pois não é possível saber com exatidão o tempo de invocação nestes casos. Entretanto, quando o tempo ocupado de uma mensagem é igual a zero, sabe-se que o início de transmissão coincide com o momento de invocação da mensagem.

Para estimar o período das mensagens no barramento é necessário usar os casos onde exista certeza do tempo de invocação das mensagens, onde o tempo ocupado seja zero.

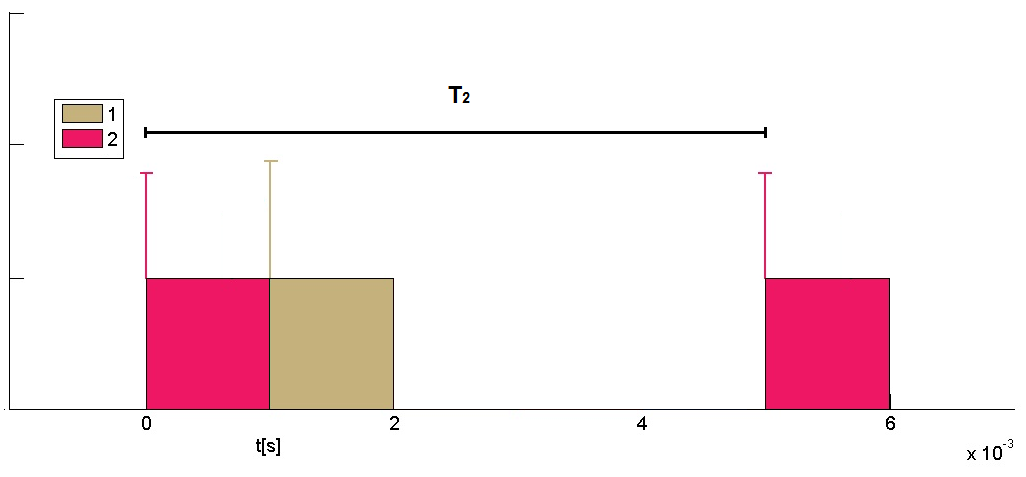


Figura 49 – período da mensagem 2.

Fonte: Autoria Própria

A figura acima mostra duas invocações da mensagem 2 que podem ser usadas para calcular o seu período de invocação, pois elas não apresentam tempo ocupado.

### Método

O método proposto se utiliza do parâmetro de tempo ocupado de cada invocação para encontrar estimativas do tempo de resposta de cada mensagem e então formar uma curva de probabilidade do tempo de resposta.

Inicialmente, para explicar o método proposto de forma sucinta, será utilizado um exemplo com uma mensagem aperiódica, onde ela é invocada dependendo de eventos não previsíveis. Como as mensagens aperiódicas não apresentam período, os valores encontrados de tempo de resposta para os casos onde exista tempo ocupado terão uma incerteza maior que mensagens periódicas, pois não existe indício de quando a mensagem foi invocada.

#### Relação entre o tempo ocupado e a probabilidade do tempo de resposta

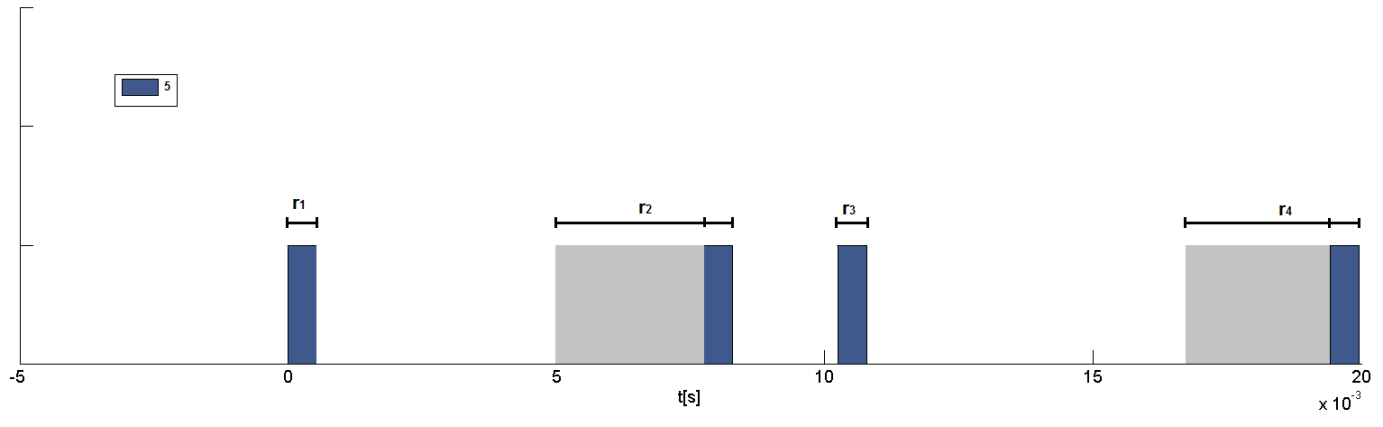


Figura 50 – Tempos de resposta.

Fonte: Autoria própria.

Através da Figura 50 pode-se observar que para calcular o tempo de resposta das instâncias sem tempo ocupado (r1 e r3) é possível utilizar a equação (8). Sabendo-se que o tempo de espera (*q)* desta instância é zero, tem-se que o valor do tempo de resposta para mensagens sem tempo ocupado é igual ao tempo de transmissão desta instância ().

Já as mensagens que apresentam tempo ocupado terão uma faixa de tempos de resposta. Estes compreendem o intervalo entre o tempo de transmissão da mensagem e a soma do tempo de transmissão com o tempo ocupado.

Cada instância da mensagem será considerada como um evento (), pertencente ao conjunto de todos os eventos medidos desta mensagem (Ω). ⊂ Ω. Ω = {}. O conjunto de todos os eventos medidos (Ω) está contido no conjunto universo de eventos possíveis (U). Ω ⊆ U. É possível considerar o conjunto de medições igual ao conjunto universo se for usado um tempo, de *log*, grande o suficiente na análise, para que sejam vistos todos os eventos possíveis no *log*.

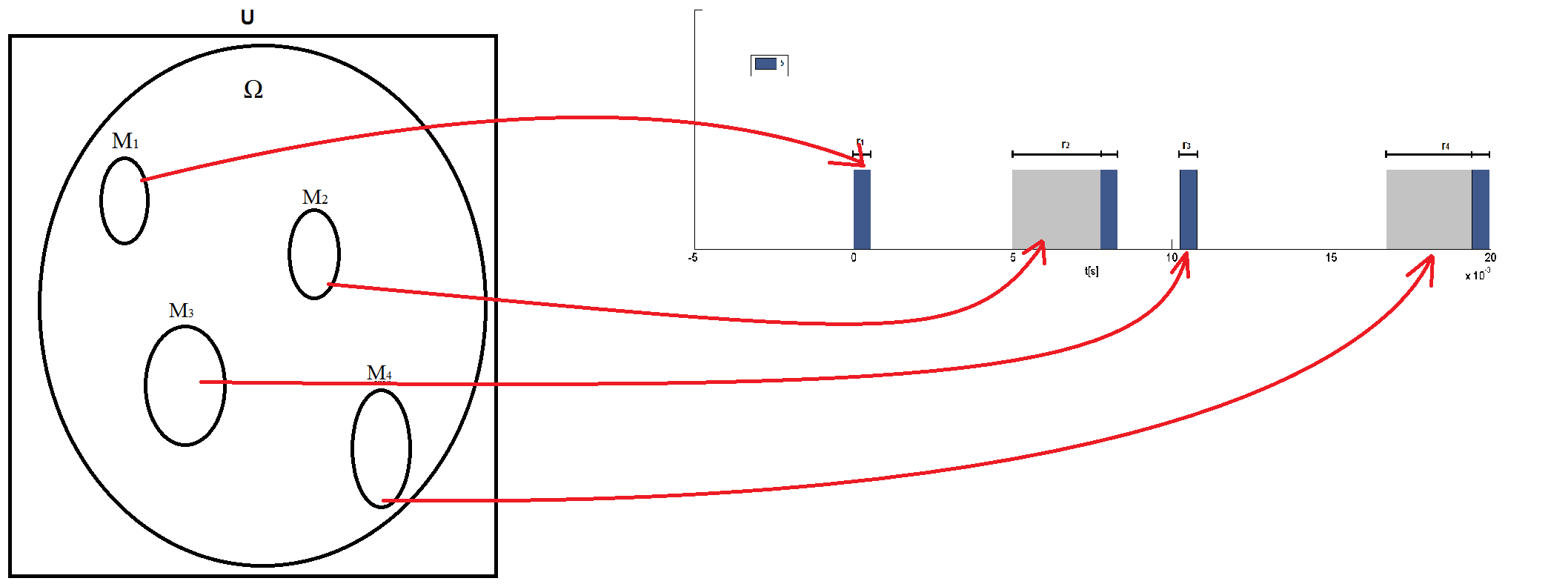


Figura 51 – Conjunto de eventos medidos.

Fonte: Autoria própria.

Os eventos serão considerados mutuamente exclusivos, onde seus conjuntos não apresentam intersecção, pois somente um evento pode acontecer por vez. Por exemplo, se uma instância não contém tempo ocupado ela não pode ter um tempo ocioso ao mesmo tempo.

A probabilidade do tempo de resposta (r) ser menor que um determinado tempo (t), pode então ser calculada com as relações que se seguem:

|  |  |  |
| --- | --- | --- |
|  |  | (71) |
|  |  | (72) |
|  |  | (73) |

Como os eventos da mensagem são eventos disjuntos, pode se considerar que a probabilidade das uniões é a soma das probabilidades em cada evento.

|  |  |  |
| --- | --- | --- |
|  |  | (74) |

Utilizando o seguinte axioma da probabilidade condicional, e aplicando em (74), tem-se (76):

|  |  |  |
| --- | --- | --- |
|  |  | (75) |
|  |  | (76) |

A probabilidade de um evento ocorrer é a razão entre o número de vezes que o evento ocorreu () pelo número de eventos observados (N), a equação (76) pode ser simplificada para:

|  |  |  |
| --- | --- | --- |
|  |  | (77) |

|  |  |  |
| --- | --- | --- |
|  |  | (78) |

Observa-se que um evento que não apresenta tempo ocupado também constitui em um evento e deve ser considerado na análise de probabilidade.

Como o resultado final desejado é uma função distribuição acumulada (CDF) dos tempos de resposta, serão usadas as densidades de probabilidade invocação de cada instância, criando uma função densidade de probabilidade de todas as invocações, denominada .

Para obter a distribuição acumulada do tempo de resposta de um evento , utilizamos a sua função densidade de probabilidade do seu tempo de resposta , a qual é obtida em função da probabilidade do seu tempo de invocação, .

|  |  |  |
| --- | --- | --- |
|  |  | (79) |
|  |  | (80) |

Através da equação do tempo de resposta (81), podemos encontrar os limites para avaliar a relação de probabilidade entre as *pdfs* (função densidade de probabilidade) de invocação e resposta.

|  |  |  |
| --- | --- | --- |
|  |  | (81) |
|  |  | (82) |
|  |  | (83) |

Uma vez possuindo a relação entre as probabilidades de invocação com o tempo de resposta de cada evento , deve-se fornecer um meio de agrupá-los, a fim de obter uma função de densidade de probabilidade de todo o conjunto de medições , o qual se aproximará do conjunto ideal (universo ).

Para obter a função de probabilidade acumulada de basta integrá-la.

Juntando a equação (77) com a relação obtida (83), e incluindo a função densidade de probabilidade de invocação , tem-se que:

|  |  |  |
| --- | --- | --- |
|  |  | (84) |
|  |  | (85) |
|  |  | (86) |
|  |  | (87) |
|  |  | (88) |
|  |  | (89) |
|  |  | (90) |
|  |  | (91) |
|  |  | (92) |

A função corresponde à pdf do tempo de resposta individual de cada invocação em função da pdf do tempo de invocação , as quais são somadas e ponderadas a fim de gerar a função global de tempo de resposta .

#### Modelo de PDF para mensagens aperiódicas

As mensagens aperiódicas com tempo ocupado () apresentam funções de densidade de probabilidade () uniforme (iniciadas em e terminadas em ), onde todos os valores de tempo de invocação dentro do intervalo são equiprováveis, então a equação (89) gera as seguintes equações:

|  |  |  |
| --- | --- | --- |
|  |  | (93) |
|  |  | (94) |
|  |  | (95) |

#### Modelo de PDF para mensagens sem tempo ocupado

A contribuição das mensagens que não tem tempo ocupado ( também precisam ser levadas em consideração ao computar a probabilidade do tempo de resposta, assim, a função para esses casos pode ser modelada por um delta de dirac no seu tempo de invocação .

|  |  |  |
| --- | --- | --- |
|  |  | (96) |
|  |  | (97) |
|  |  | (98) |

Estas equações funcionam bem para mensagens aperiódicas, porém, as mensagens periódicas podem fornecer uma informação a mais, o período, e com ele é possível determinar o momento de invocação da mensagem no tempo ocupado e assim determinar com mais precisão o tempo de resposta.

Ainda que as mensagens periódicas apresentem melhora no método para achar os tempos de resposta, elas ainda têm problemas. O método para encontrar o período das mensagens periódicas e os seus problemas são apresentados nas sessões seguintes.

#### Exemplo de aplicação em uma mensagem não periódica

Para fazer a análise de tempos de resposta para uma mensagem aperiódica, basta aplicar as equações desenvolvidas anteriormente, a fim de se obter a pdf total.

Tabela 10- Exemplo de transmissão com tempo ocupado

|  |  |  |
| --- | --- | --- |
| Invocação(i) | Início de tempo ocupado (bi)[s] | Início de transmissão (si)[s] |
| 1 | 0 | 0,001034 |
| 2 | 0,002336 | 0,004104 |
| 3 | 0,005743 | 0,007653 |
| 4 | 0,009629 | 0,01147 |
| 5 | 0,012563 | 0,013798 |
| 6 | 0,015184 | 0,017116 |
| 7 | 0,018985 | 0,020364 |
| 8 | 0,021376 | 0,022981 |
| 9 | 0,024883 | 0,025968 |
| 10 | 0,027233 | 0,028519 |

Assim, utilizando os dados fornecidos na Tabela 10, que corresponde à transmissão de 10 instâncias de uma mensagem aperiódica, utiliza-se a equação (95), somando-se todos os e dividindo-os pelo número total de instâncias, uma vez que cada instância corresponde a um evento diferente.

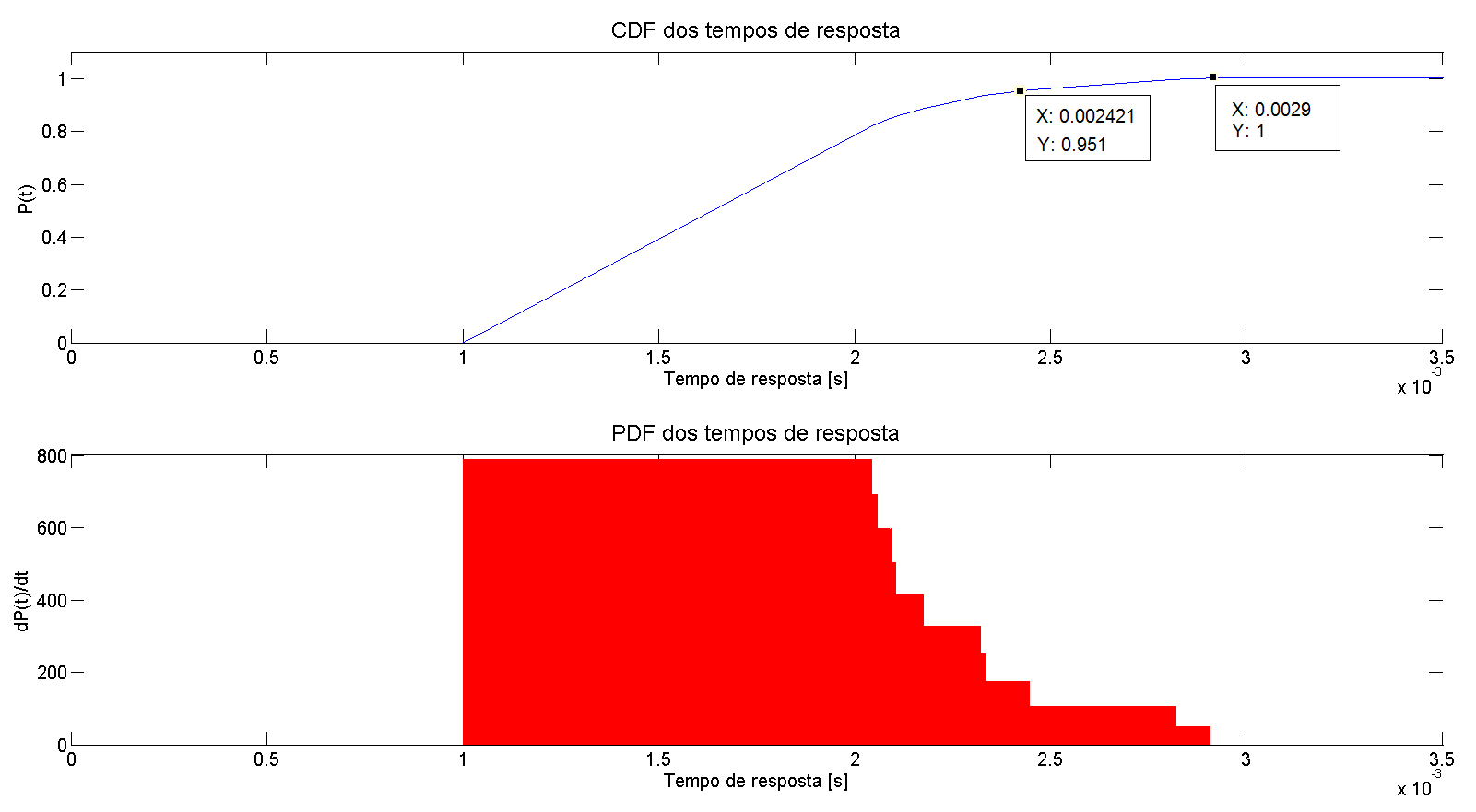


Figura 52- CDF e PDF para os dados da Tabela 10

Fonte: Autoria Própria

Como resultado temos a Figura 52, onde se pode ver que em 95% das instâncias o tempo de resposta é menor que 2.4ms, e em 100% das instâncias, ele é inferior a 2.9ms.

Pela análise ter sido realizada para uma mensagem aperiódica, a pdf utilizada como base dentro do tempo ocupado foi uma distribuição uniforme.

Nota-se que, se a análise de pior caso através de equações fosse utilizada, seria necessário ter conhecimento prévio sobre o sistema, como o período e tempo de transmissão de cada mensagem mais prioritária, e, por se tratar de uma mensagem aperiódica, também é necessário o período mínimo entre invocações sucessivas da mensagem que está sendo colocada sob análise.

Essas necessidades tem vários impactos no momento em que se realiza esse tipo de análise, e, além disso, resultaria em um valor bastante pessimista de tempo de resposta, sem fornecer uma cdf.

### Estimativa do período

Como já mencionado anteriormente é necessário que as instâncias usadas para estimar o período de uma mensagem sejam as instâncias com tempo ocupado igual a zero.

A diferença do tempo inicial de transmissão entre duas mensagens pode ser usada para calcular o período da mensagem, mas devem ser usadas duas instâncias seguidas, com tempo ocupado zero, para poder calcular o período com uma incerteza menor.

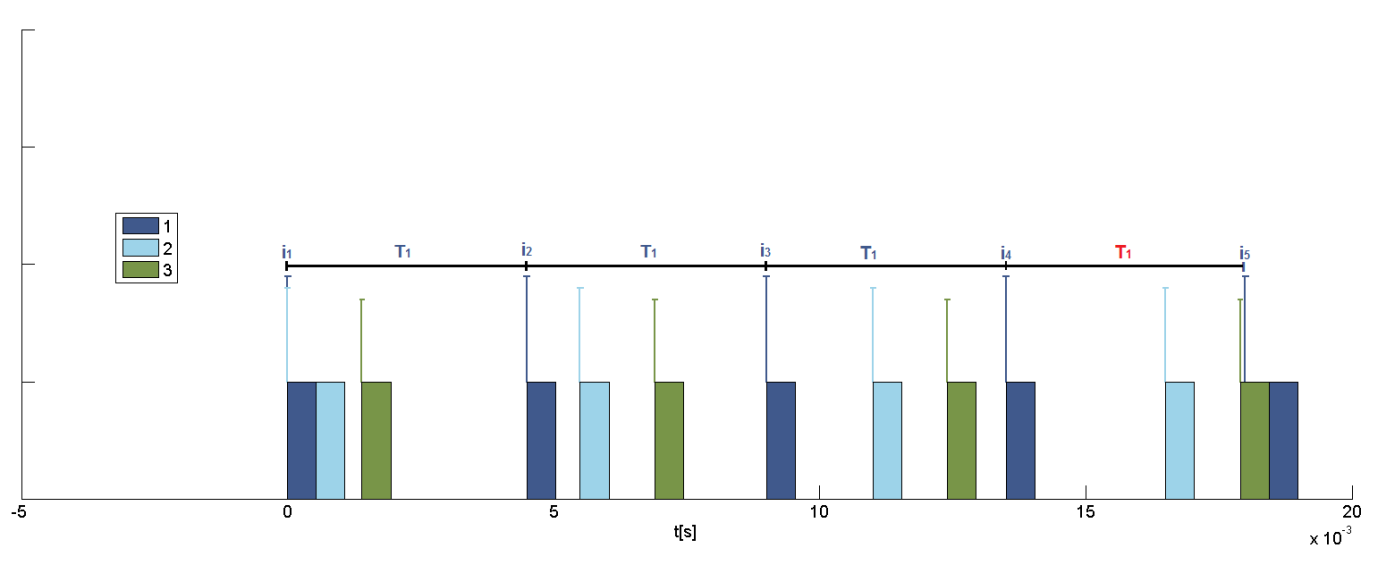


Figura 53 – Período mensagem 1.

Fonte: Autoria Própria

A figura acima mostra um conjunto de mensagens periódicas, onde o período da mensagem 1 () pode ser obtido da diferença entre dois inicios de transmissão consecutivos quaisquer, a não ser da última invocação () que apresenta período ocupado. Em um barramento ideal seria possível saber o tempo de invocação da ultima invocação projetando a penúltima invocação para frente usando um período, mas em barramentos reais isto não é possível.

Essa relação só pode ser usada em um ambiente ideal, onde as mensagens apresentam período constante e sem *jitter*. No barramento real as mensagens apresentam comportamentos inconstantes de períodos, mensagens sem período e nível de *jitter* alto.

Para obter o período em uma rede real é necessário então obter o valor médio das diferenças entre as invocações seguidas com tempo ocupado zero, para que se possa fazer uma projeção dos tempos de invocação nos tempos ocupados e obter um tempo de resposta mais preciso para mensagens periódicas.

É possível montar uma PDF para cada invocação da mensagem, com as diferenças de tempo de invocação, para serem usadas nas CDF dos tempos de resposta. Essa PDF vai representar também o nível de *jitter* da mensagem.

### Jitter

Em uma rede com um nível de jitter muito grande as mensagens apresentam um valor de período diferente do valor originalmente programado. Cada invocação das mensagens vai apresentar uma diferença entre o tempo de invocação real e o tempo de invocação do módulo, o qual é o tempo em que o módulo decide fazer com que a mensagem participe do processo de arbitragem.

A Figura 54 apresenta uma mensagem periódica, onde cada invocação está separada da outra pelo período () e por um jitter (). O jitter é sempre um valor positivo, que se adiciona ao período da mensagem.

É possível separar as fontes de jitter do sistema de um barramento CAN em duas principais, sendo elas o tempo necessário para a tarefa do controlador gerar a mensagem e o tempo da mensagem na fila de hardware de CAN do controlador.

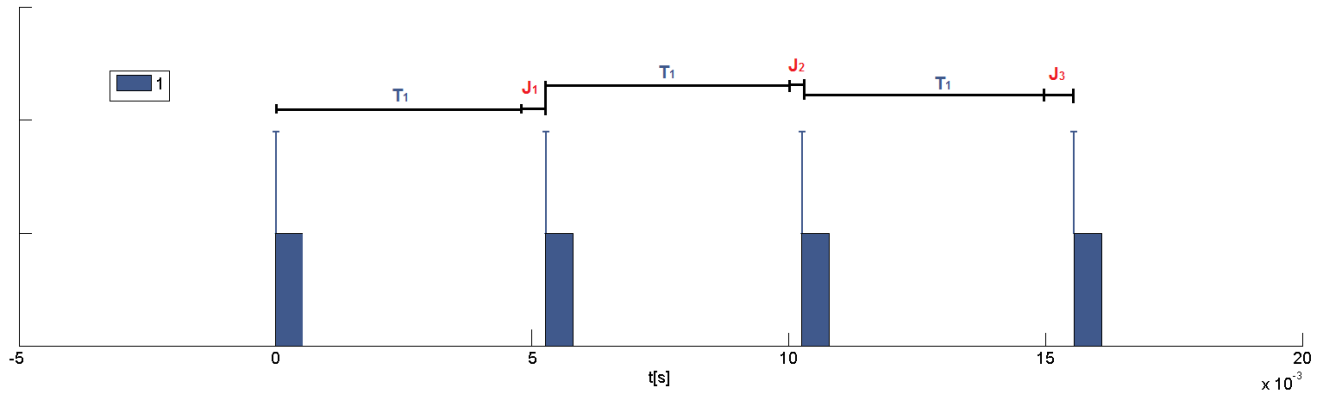


Figura 54 – Invocações com jitter.

Fonte: Autoria própria.

Geralmente um controlador em um barramento CAN contém algumas tarefas que geram informações que precisam ser enviadas a outros controladores através do barramento CAN. A tarefa geradora das mensagens pode sofrer algumas preempções, ou interrupções, enquanto processa os dados para gerar as mensagens e este tempo em que a tarefa foi interrompida será adicionado ao período da mensagem para próxima instância, gerando o jitter aqui chamado de jitter de período, .

Os controladores de hardware CAN integrados nos controladores do sistema, geralmente possuem um buffer de mensagens a serem transmitidas no barramento. Este buffer é preenchido pelo *software* à medida que as mensagens são geradas, o tempo de passagem das mensagens do *software* para o *hardware* é variável, dependendo da implementação do *software*. As mensagens podem apresentar atrasos devido ao tamanho do buffer do hardware, se há inversão de prioridades. Esse jitter é aqui chamado de jitter de fila, .

Devido a diferença de implementação entre ECUs, é difícil generalizar os motivos dos jitter de fila e de período. No entanto, considerando-se os dois tipos de erro toma-se uma abordagem um pouco mais genérica, que nos permite separar casos em que um dos valores é maior que o outro, levando a resultados mais próximos da realidade.

#### Modelo de erro

Existem diversos tipos de escalonamento a serem utilizados nos sistemas operacionais presentes em uma ECU, algo que influencia diretamente no jitter de invocação de mensagens periódicas, uma vez que as mesmas acabam por herdar o comportamento das tarefas que as invocam.

Uma implementação mais simples de tarefa periódica, como fazer com que a mesma “durma” por um tempo *T* após o fim de sua execução, irá resultar em um período de execução diferente de *T*, devido ao tempo em que a tarefa demora desde que iniciou a sua execução, até que a tenha finalizado.

No modelo de erro proposto, esse tipo de caso é considerado, pois o incremento de período é transmitido à mensagem a ser invocada.

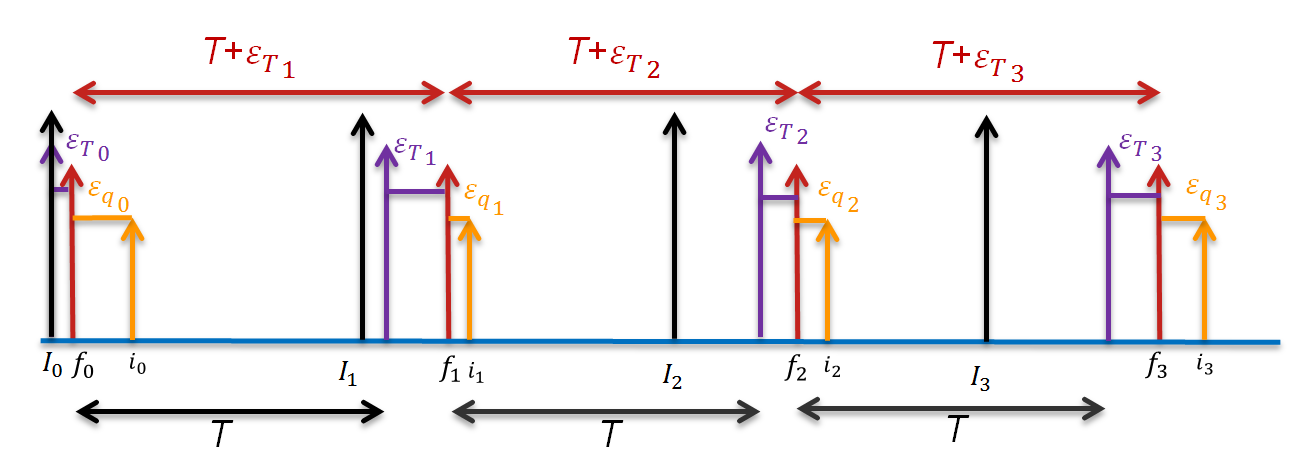


Figura 55 - Modelo de erro de período e de fila

Fonte: Autoria Própria

A Figura 55 representa o modelo de erro utilizado, onde as variáveis Ik, fk,ik, consistem no tempo de invocação sem erro, com erro de período e com erro de período e fila, respectivamente.

|  |  |  |
| --- | --- | --- |
|  |  | (99) |
|  |  | (100) |

As equações (99) e (140) representam as variáveis correspondentes a primeira invocação, enquanto que a (101) e (102) correspondem a segunda.

|  |  |  |
| --- | --- | --- |
|  |  | (101) |
|  |  | (102) |

Repetindo o processo anterior para uma instância *k*, tem-se que:

|  |  |  |
| --- | --- | --- |
|  |  | (103) |
|  |  | (104) |
|  |  | (105) |

Através da equação (105) pode-se observar que o tempo de invocação real vai se distanciando cada vez mais (em função de *k*) do seu tempo ideal de invocação. Assim, dificulta-se a projeção do tempo de invocação estimado para a obtenção da pdf do tempo de resposta.

Existe a necessidade de se reavaliar o período com o novo modelo de erro, pois o erro de período introduzirá incertezas no mesmo.

#### Período

O período de uma mensagem periódica é definido como a diferença entre duas invocações sucessivas, porém, como temos incertezas associadas às invocações, trataremos o período como a *média* das diferenças entre invocações sucessivas.

Assim, utilizando a equação (105), tem-se que:

|  |  |  |
| --- | --- | --- |
|  |  | (106) |
|  |  | (107) |
|  |  | (108) |
|  |  | (109) |

O erro de período tem um valor de média , o qual pode ser removido do somatório e colocado em função da distância entre *k* e *g*, como segue abaixo:

|  |  |  |
| --- | --- | --- |
|  |  | (110) |
|  |  | (111) |
|  |  | (112) |
|  |  | (113) |

A equação (109) nos fornece a relação no tempo entre duas invocações, sucessivas ou não. Ela será utilizada para encontrar o período (média entre duas invocações sucessivas), portanto, *k=g+1*.

|  |  |  |
| --- | --- | --- |
|  |  | (114) |
|  |  | (115) |
|  |  | (116) |
|  |  | (117) |
|  |  | (118) |
|  |  | (119) |

Assim, a fim de obter o valor através do log de mensagens, deve-se considerar apenas mensagens sucessivas sem tempo ocupado, os quais apresentam tempos de início de execução equivalentes aos tempos de invocação.

#### PDF do Jitter através de invocações sucessivas

Na equação da diferença entre duas invocações (115), existe um termo aleatório que depende dos erros de fila e de período, como apresentado na seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (120) |

Esse erro consiste na diferença entre erros de fila entre duas invocações somado com o erro de período da nova invocação, o que altera o valor teórico entre invocações, que seria de *T*.

O termo é subtraído dessa variável e somado ao período para que a média de seja igualada a zero, simplificando alguns cálculos.

|  |  |  |
| --- | --- | --- |
|  |  | (121) |

A nova equação (121) nos fornece um meio de estimar a próxima invocação de uma mensagem sabendo a invocação de sua instância anterior, como, por exemplo, se supormos que não existe erro de período e de fase, a equação resulta em (122), que nos fornece o tempo de *g+1* sabendo-se o tempo de *g*.

|  |  |  |
| --- | --- | --- |
|  |  | (122) |

Como mencionado anteriormente, essa equação seria ideal para o problema a ser resolvido se não houvesse jitter, uma vez que duas instâncias sem período ocupado nos forneceriam o seu tempo de invocação e período através do log, possibilitando a estimativa dos restantes. Para uma análise mais coerente, todavia, tem-se a necessidade de medir os erros do barramento e utilizá-los para a estimativa dos tempos de invocação.

A equação (119) nos fornece um meio para obter o valor do período , o que nos leva a equação (124), a qual nos fornece os valores de erro para invocações sucessivas sem tempo ocupado, possibilitando a estimativa de uma pdf da variável (jitter), através de amostras de invocações sucessivas.

|  |  |  |
| --- | --- | --- |
|  |  | (123) |
|  |  | (124) |

#### PDF do Jitter através de invocações distantes

A estimativa da pdf da variável é um pouco diferente se invocações que não são sucessivas () forem utilizadas como amostras, pois o erro de período se soma, como demonstrado na equação (113).

|  |  |  |
| --- | --- | --- |
|  |  | (125) |
|  |  | (126) |
|  |  | (127) |

A soma de duas variáveis aleatórias independentes, , equivale a convolução de sua pdf :

|  |  |  |
| --- | --- | --- |
|  |  | (128) |

Assim, a pdf de , corresponde a equação abaixo, onde a pdf de é convoluída (operador de convolução - ) *k-g*  vezes, devido ao somatório.

|  |  |  |
| --- | --- | --- |
|  |  | (129) |

Supondo que o erro tenha a sua pdf modelada por um delta de dirac com média , não existe contribuição do mesmo na pdf de pois a convolução de uma função com um delta é a própria função, levando à seguinte equação:

|  |  |  |
| --- | --- | --- |
|  |  | (130) |

Essa equação demonstra que quando o erro de período é modelado como um delta (variância muito menor que a do erro de jitter), o jitter total não depende da distância entre as invocações, possibilitando o seu uso como amostras na estimativa da pdf total.

Assim, a utilização de mensagens não sucessivas na estimativa deve ser avaliada de acordo com a variância do erro de período daquela mensagem, assunto que será abordado no próximo tópico.

#### Variância do erro de período

A variância do erro de período tem importância no momento de estimar se o erro do período é desprezível em relação ao erro de fila, possibilitando a utilização, como amostra, de qualquer invocação que não tem tempo ocupado.

Dividindo a equação (126) por *k-g*, tem-se:

|  |  |  |
| --- | --- | --- |
|  |  | (131) |
|  |  | (132) |

Utilizando valores de *k-g* maiores que 30, o erro de fila torna-se muito menor que o erro acumulado de período, tornando sua variância desprezível, enquanto que o somatório do erro de período segue o teorema do limite central, tendendo a uma gaussiana com a seguinte variância:

|  |  |  |
| --- | --- | --- |
|  |  | (133) |

Portanto, a variância estimada de é:

|  |  |  |
| --- | --- | --- |
|  |  | (134) |

Onde o termo é dado pela variância do termo do lado esquerdo da equação (131), levando a:

|  |  |  |
| --- | --- | --- |
|  |  | (135) |

Observa-se que através deste método, somente obteve-se a variância de , o que apenas fornece uma ideia de quão espalhada é a sua pdf, e não o seu formato.

A fim de decidir se o erro de período é desprezível, deve-se comparar com a variância da distribuição de :

|  |  |  |
| --- | --- | --- |
|  |  | (136) |
|  |  | (137) |
|  |  | (138) |

Uma vez que o valor de e foram encontrados, a sua razão será comparada a um limiar que nos informará se o erro de período é desprezível.

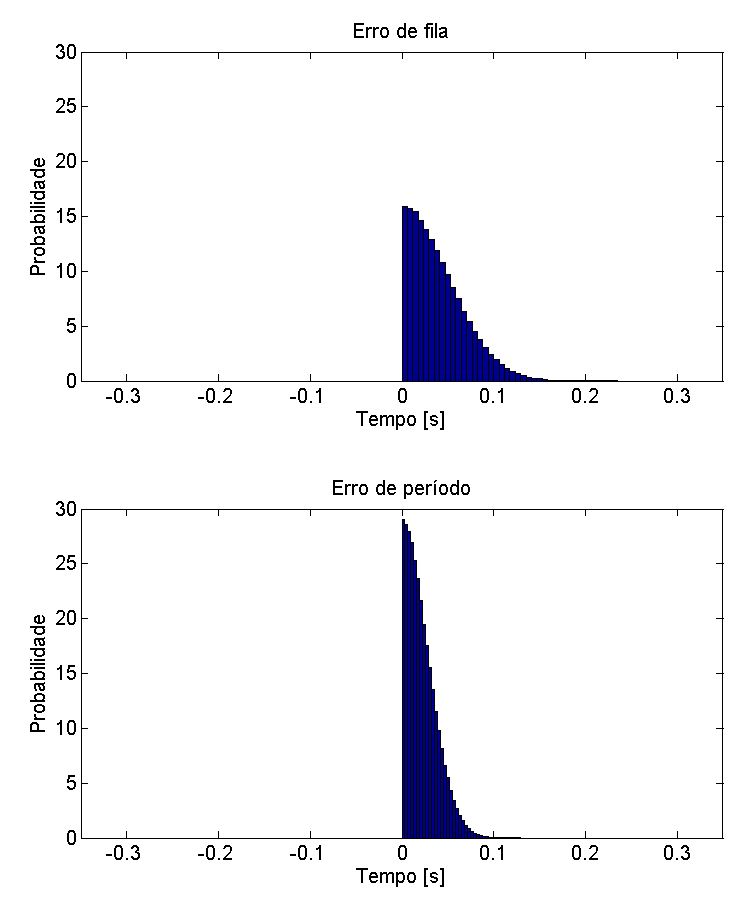


Figura 56 - Erro de período e erro de fila exemplificados como gaussianas dobradas

Fonte: Autoria Própria

A fim de testar a estimativa de variância de cada erro, um código no matlab foi gerado, onde foram simuladas diversas invocações de uma mensagem através da equação (121). As pdfs utilizadas para gerar os erros de fila e de período foram gaussianas dobradas (*folded normal distribution*), como representado na Figura 56, e com respectiva variância de 9.08e-04 e 2.72e-04, correspondendo a uma razão entre elas de 30%.

Ao estimar o erro de instâncias consecutivas, quem predomina é o erro de fila, e quando trata-se de instâncias distantes (distância maior que 30), quem predomina é o erro de período. Para isso, o limiar superior para utilizar como amostras de erro é a distâncias de 30, pois o formato da pdf já está totalmente alterado e tende a uma gaussiana, independente das distribuições envolventes (teorema do limite central), porém, há a necessidade de se encontrar um limiar menor que esse, pois procura-se que a soma do erro de período não seja predominante.

Para verificar se o erro de período (e sua soma) é desprezível em relação ao de fila, obtêm-se a curva de erro total, Figura 57, causada pelo conjunto dos erros, estima-se a variância de cada erro, a partir das equações (135) e (138), e então compara-se a razão entre os resultados obtidos com um limiar que determina a desprezibilidade do erro de período.

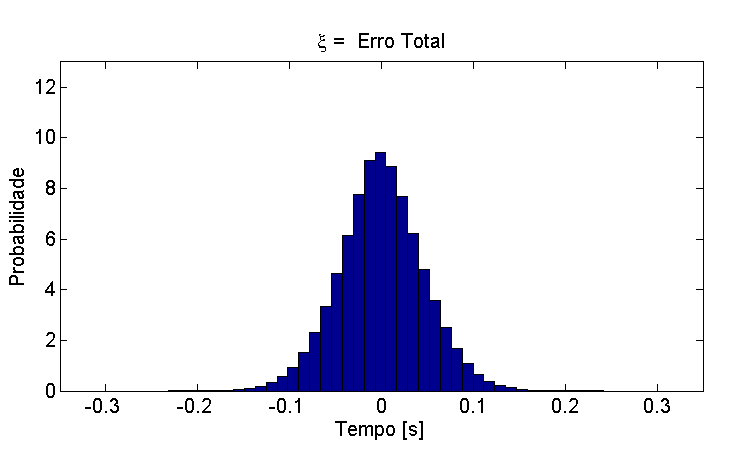


Figura 57 - Erro gerado pelas imagens da Figura 56 através da equação (120) para instâncias consecutivas

Fonte: Autoria Própria

As variâncias estimadas de erro de fila e de período, foram, respectivamente, 8.90e-04 e 3.03e-04. Esses valores foram obtidos utilizando um valor de *k-g* de 30 e 10000 amostras, algo que, com um log suficientemente grande, é fácil de ser obtido para as mensagens mais prioritárias.

Existem erros no método de estimativa das variâncias devido ao valor de *k-g* utilizado (quanto maior melhor) e o número de amostras. Assim, a diferença percentual entre as variâncias reais e as estimadas, de fila e de período, foram respectivamente, 2.66% e 10.5%, e a razão entre variâncias foi de 34% (30% era o valor real).

Os erros de estimativa são aceitáveis para essa aplicação, e um limiar de razão entre variâncias deve ser estipulado levando em consideração que esse tipo de incerteza existe.

Determina-se que a parcela de variância proveniente do erro de fila e de período devem ser, no máximo, iguais, assim, pode-se calcular um limite para distância entre instâncias. Observa-se que o limite máximo ainda é de *k-g=30*, pois, o formato da pdf resultante se altera demasiado.

|  |  |  |
| --- | --- | --- |
|  |  | (139) |
|  |  | (140) |
|  |  | (141) |

Ou seja, se a razão entre os erros (fila sobre período) for de 10%, pode-se utilizar um (*k-g)* de até 20.

As amostras de valores de erro (lado direito da equação (126) (k-g) ) valem como uma estimativa de somente se o *k-g* utilizado for menor que (*k-g)máx* e que 30, o que leva finalmente a:

|  |  |  |
| --- | --- | --- |
|  |  | (142) |

#### Projeção da pdf de tempo de invocação

Através dos resultados anteriores, pode-se observar que através da média das diferenças de invocações sucessivas, obtemos o valor , o qual é utilizado na seguinte equação para projetar os possíveis tempos de invocação.

|  |  |  |
| --- | --- | --- |
|  |  | (143) |

A projeção desses tempos consiste em basear-se em uma mensagem que foi transmitida sozinha (tempo ocupado igual a zero) e através do conhecimento da pdf do jitter e do seu período médio, estimar a função de probabilidade correspondente ao seu tempo de invocação, que, como visto anteriormente, será utilizado para calcular o tempo de resposta da mensagem como um todo.

Assim, por exemplo, conhecendo um tempo de invocação *i6*, e observando que i7,i8,i9 e i10 tem período ocupado, as seguintes equações correspondem à invocação dessas mensagens:

|  |  |  |
| --- | --- | --- |
|  |  | (144) |
|  |  | (145) |
|  |  | (146) |
|  |  | (147) |

Essas equações apresentam as variáveis aleatórias e independentes e , que se agrupam no sistema e resultam em uma pdf diferente para cada (*k-g*), pois o erro de período vai se somando, resultando na convolução sucessiva de uma pdf com a pdf do erro de período, e graficamente representado na Figura 58.

Nessa figura estão ilustrados também os tempos ocupados de cada invocação, os quais serão utilizados posteriormente para melhorar a análise de tempo de resposta.

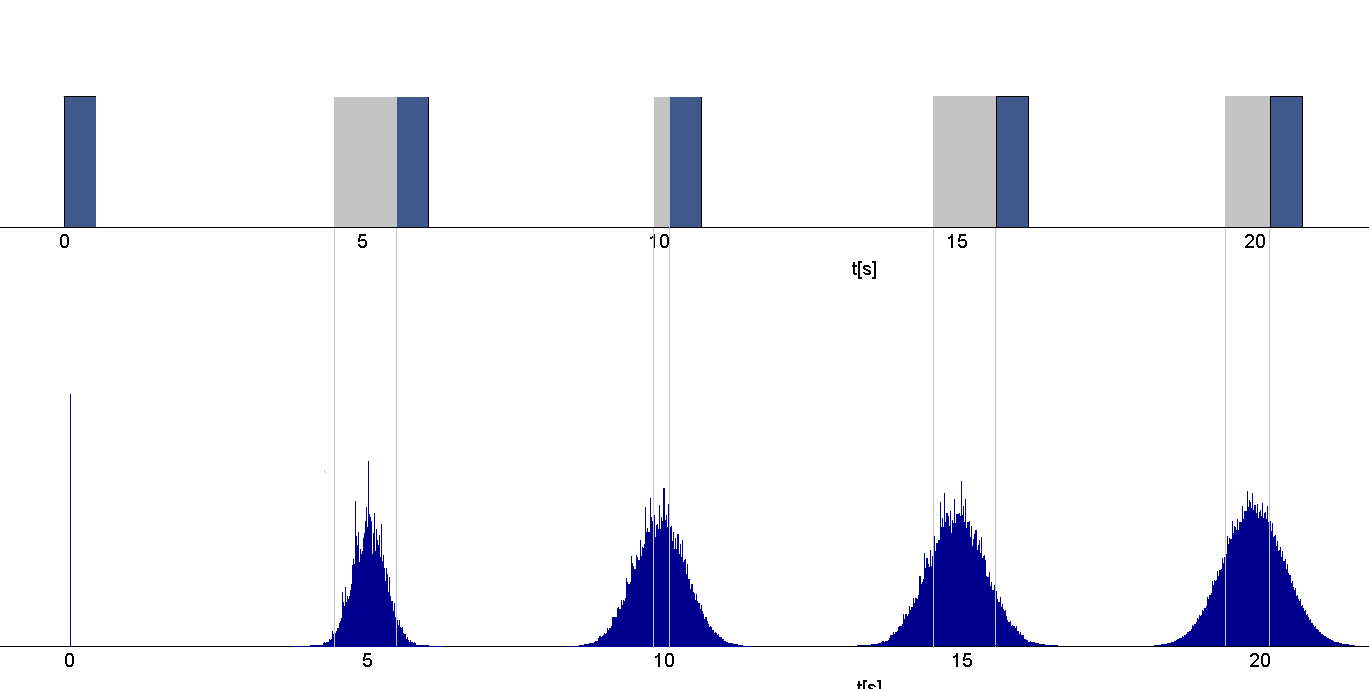


Figura 58 - Pdf ideal do tempo de invocação onde o erro de período vai se somando

Fonte: Autoria Própria

Se a pdf do erro de período e de fila fossem previamente conhecidas, a análise poderia estimar a pdf de qualquer invocação que ocorresse a partir de uma única mensagem que fosse transmitida sem período ocupado. Existe, porém, uma grande dificuldade em diferenciar que parte do erro é de fila e qual é de período através de medições.

Portanto, assim como feito nas seções 6.4.4.4 e 6.4.4.5, através do método de comparação de variância apresentado nas equações (135), (138) e (141), apenas a pdf será estimada, a qual ignora a existência da distribuição erro de período para invocações com distância até (*k-g)máx*.

A seguinte aproximação foi feita porque anteriormente determinou-se que para um *k-g* menor que (*k-g)máx* o erro de período era desprezível em relação ao de fila.

|  |  |  |
| --- | --- | --- |
|  |  | (148) |
|  |  | (149) |
|  |  | (150) |
|  |  | (151) |

Assim, a pdf de invocação de cada mensagem será a distribuição porém com média igual a (k-g)(μT- μTε), porém, quando (*k-g*) se torna maior que o seu valor máximo, deve-se aplicar uma abordagem um pouco diferente.

#### Tempo ocupado na estimativa da invocação de uma mensagem periódica

O significado matemático de quando o erro de período já não se torna desprezível em relação ao de fila é, que devido ao teorema do limite central, a pdf de invocação tende a uma gaussiana com uma variância diretamente proporcional a (*k-g*), como demonstrado anteriormente*.*

Para contornar esse problema, poderia ser utilizada uma gaussiana com variância (*k-g*)σεT a partir de (*k-g*)>(*k-g*)max . Há, porém, outra abordagem envolvendo o tempo ocupado de cada mensagem pode ser utilizado.

Como definido anteriormente, o tempo ocupado de uma mensagem é a janela (intervalo) em que não se tem certeza sobre a invocação da mensagem, porém, têm-se certeza de que ele ocorreu dentro da janela. Isso leva à conclusão de que na verdade existe zero probabilidade de invocação fora do tempo ocupado, e uma vez que a gaussiana com variância (*k-g*)σεT tem a inclinação de sua curva desprezível dentro do tempo ocupado, devido ao fato de sua variância ser muito maior que a largura dessa janela, podemos utilizar uma pdf com distribuição uniforme, igualmente ao que foi feito às mensagem aperiódica (Figura 59).

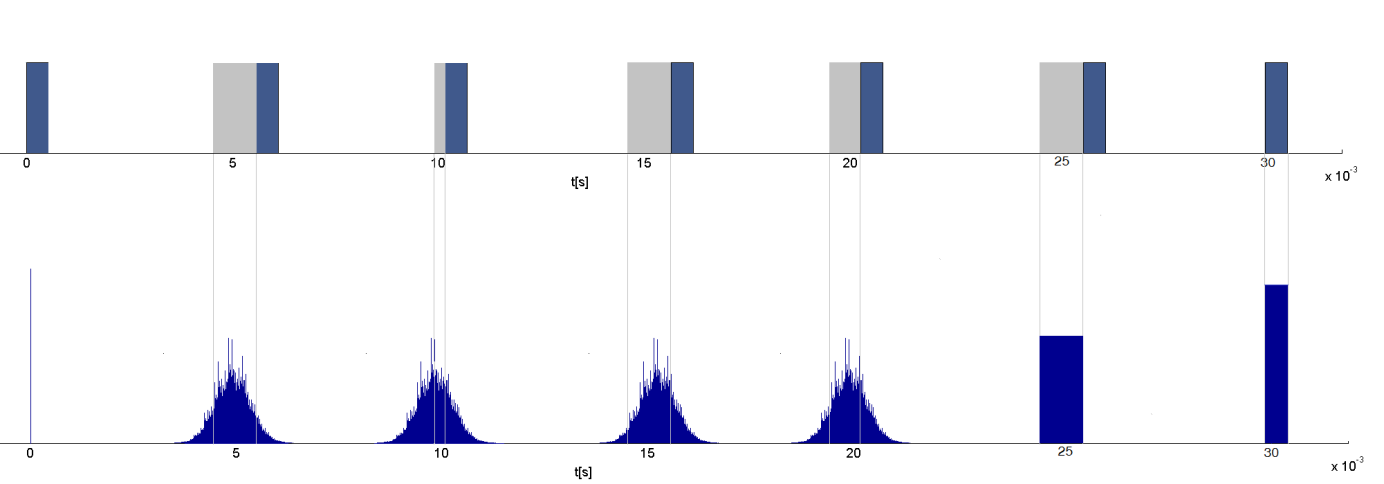


Figura 59 - Projeção dos tempos de invocação sem considerar erro de período até a quinta invocação

Fonte: Autoria Própria

Expandindo essa abordagem, pode-se também eliminar a parte da pdf de invocação que está fora do tempo ocupado de todas as mensagens, já que existe a certeza de que as invocações não ocorreram ali, levando a uma análise mais realista.

Observa-se também que ao eliminar a parte da pdf, ela deve ser normalizada, pois a sua área deixa de ser igual à unidade. Isto não faz sentido, uma vez que o fato de sua área ser um dentro do tempo ocupado significa justamente que a mensagem foi invocada dentro daquele intervalo.

Assim, o resultado dos tempos de invocação da Figura 59 É representado na Figura 60.

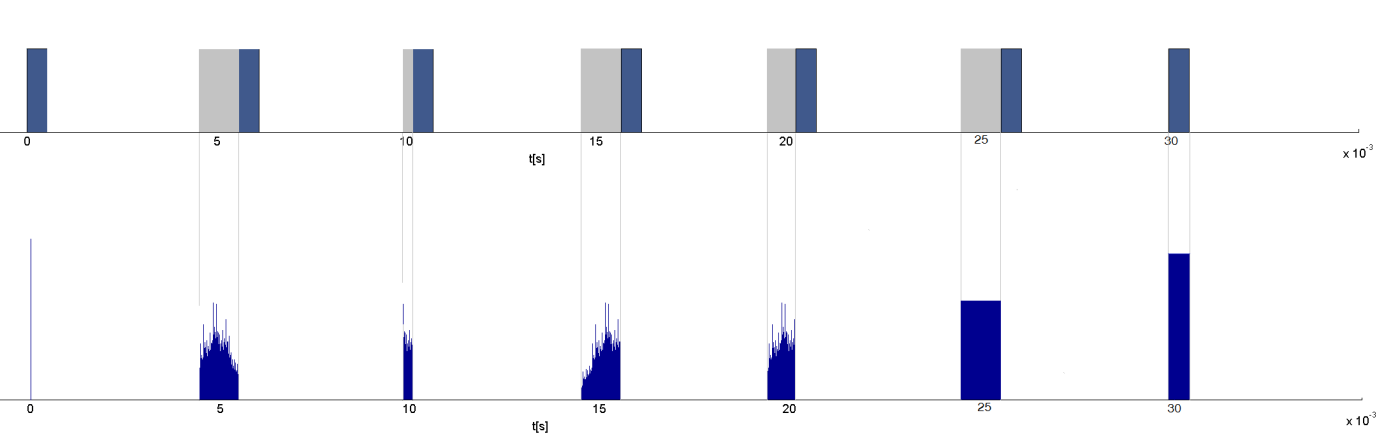


Figura 60 - Tempo de invocação restritos pelo tempo ocupado

Fonte: Autoria Própria

Levando a equação (143) para o seu domínio de probabilidade, obtemos a pdf dos tempos de invocação () para (k-g) menores que o máximo, onde *g* é a última transmissão sem tempo ocupado.

|  |  |  |
| --- | --- | --- |
|  |  | (152) |

Durante a estimativa do jitter, devido a um número não suficiente de amostras, nem todos os eventos possíveis de jitter podem ter ocorrido, o que pode levar a uma intersecção nula entre a pdf do jitter e o tempo ocupado.

Nesses casos a execução é tratada como aperiódica, sendo utilizada uma distribuição uniforme restrita pelo tempo ocupado, pois não existem dados suficientes para estimar a sua invocação de uma maneira mais precisa.

### Mensagens esporadicamente periódicas

A medida em que o veículo vai sendo utilizado, algumas mensagens (periódicas e aperiódicas) param de ser utilizadas e são retomadas posteriormente. Isso não acarreta em um problema na análise quando se leva em consideração mensagens aperiódicas pois a sua probabilidade de tempo de resposta não depende de invocações anteriores, somente do tempo ocupado.

Entretanto, a análise das mensagens periódicas utiliza o tempo de invocação de uma mensagem anterior à analisada para estimar sua probabilidade de tempo de resposta, gerando a necessidade de se estimar quando uma determinada mensagem parou de ser executada e quando a mesma retornou.

Essa característica do barramento cria uma dificuldade para decidir se uma determinada mensagem é ou não periódica, pois ela pode ocorrer em rajadas (seção 6.1.7), sendo esporadicamente periódica, algo que a análise proposta deve contemplar.

Ao ser esporadicamente periódica, se apenas a variância de invocações sucessivas for levada em consideração, ela pode fornecer resultados errôneos, uma vez que a mensagem deixa de ser transmitida por alguns períodos, assim, o período interno de uma mensagem esporadicamente periódica deve ser utilizado para separar os seus momentos de execução.

O período interno da rajada deve ser previamente conhecido para a correta divisão entre as mesmas, permitindo a estimativa correta do período com jitter.

Como visto anteriormente, a seguinte equação fornece a diferença entre duas invocações sucessivas:

|  |  |  |
| --- | --- | --- |
|  |  | (153) |

A partir da equação de duas invocações sucessivas, podemos encontrar a relação entre dois inícios de transmissão em função dos seus tempos *qg+1* e *qg*, os quais correspondem aos tempos em que essas mensagens esperaram para ganhar a arbitragem do barramento.

|  |  |  |
| --- | --- | --- |
|  |  | (154) |

O valor máximo de diferença entre execuções ocorre quando a mensagem *g* não sofre nenhum tipo de atraso/jitter, porém, a mensagem *g+1* sofre os piores casos.

|  |  |  |
| --- | --- | --- |
|  |  | (155) |
|  |  | (156) |

Uma vez que *qg+1* deve ser menor que *T*, pois não há uma segurança em afirmar que todas as ECUs do veículo conseguem enfileirar sucessivas invocações de uma mesma mensagem, temos que *qg+1* ≅ *T.*

Similarmente, podemos aceitar que os piores erros de jitter sofridos pela mensagem *g+1* são pequenos em relação ao período, pois se forem demasiadamente grandes a mensagem terá uma grande variância de período e fila, sendo tratada como não periódica e não se encaixando no perfil de uma esporadicamente periódica.

Assim, a condição para que duas mensagens sucessivas estejam na mesma rajada é:

|  |  |  |
| --- | --- | --- |
|  |  | (157) |

O período não é conhecido, e a sua estimativa é realizada através da mediana da diferença de todas as invocações, pois essa operação minimiza o efeito de *outliers*  na distribuição, os quais são causados justamente pela grande diferença de tempo (maior que 2*T*) entre duas rajadas.

### Sumário do método da estimativa de pdf

Primeiramente, supõe-se que todas as mensagens são esporadicamente periódicas, fazendo-se uma tentativa de separação de rajadas. A separação entre rajadas é realizada quando a condição presente na equação (157) não for satisfeita, ou seja, a distância entre duas execuções for maior que 2*T.*

O período a ser utilizado para a separação de rajadas será a mediana da diferença entre execuções consecutivas, pois essa medida reduz a influência dos *outliers*  criados pelas diferenças de tempo entre rajadas.

Após essa separação, estima-se o período médio e a pdf do erro para mensagens consecutivas (k-g = 1) e que foram executadas sem tempo ocupado, através dos quais calculam-se as variâncias e (seção 6.4.4.5).

A razão entre as variâncias é analisada (equação (141)) e fornece um número máximo de *k-g*. Se esse número for menor que a unidade, significa que o erro de período é muito grande, e a mensagem é tratada diretamente como aperiódica; caso contrário, estima-se novamente a pdf do erro, porém, para invocações que tenham o seu *k-g*  menor que o valor máximo previamente calculado.

As mensagens são novamente separadas em grupos, onde a primeira mensagem é transmitida sozinha ou pertencente a uma nova rajada, e o restante teve período ocupado, assim, através do (*k-g)max* determina-se para cada *k-g* se a equação a ser utilizada é a do jitter medido (152) ou a do tempo ocupado (95).

Tratando-se de mensagens aperiódicas e instâncias com *k-g* maior que o (*k-g)max*, o cálculo do tempo de resposta de cada execução é feito utilizando a pdf de invocação, , como presente na equação (95).

Por outro lado, tratando-se de mensagens periódicas, a pdf do erro é projetada, como descrito na seção 6.4.4.6, e o tempo ocupado da presente invocação seleciona o pedaço da pdf ao qual aquela situação se encaixa (seção 6.4.4.7), levando à equação , cuja área necessita ser normalizada.

Uma vez que se obtenha a pdf de invocação, calcula-se a do tempo de resposta de cada execução através da equação (92), independente se a mensagem foi tratada como periódica ou aperiódica.

Após isso, somam-se as diversas pdf de tempo de resposta através da equação (91), obtendo-se a uma pdf final do tempo de resposta daquela mensagem, e consequentemente, a sua cdf, que é o resultado desejado.

# IMPLEMENTAÇÃO E RESULTADOS

O ambiente de desenvolvimento escolhido para a implementação do método proposto foi o MATLAB 2013b (*MATrix LABoratory*), o qual permite uma rápida prototipagem de algoritmos, com uma amigável interface gráfica, permitindo um fácil desenvolvimento e depuração, com uma performance de velocidade razoável.

A empresa MathWorks, produtora do MATLAB, comercializa uma série de pacotes de funções e aplicações (*toolbox*) que funcionam sobre o mesmo e facilitam o processo de desenvolvimento, porém, se algumas dessas *toolbox* fosse utilizada na implementação do método, haveria a necessidade de possuir esse conjunto de pacotes para executar a análise, portanto, foi optado por não utilizar funções extras ao MATLAB nativo.

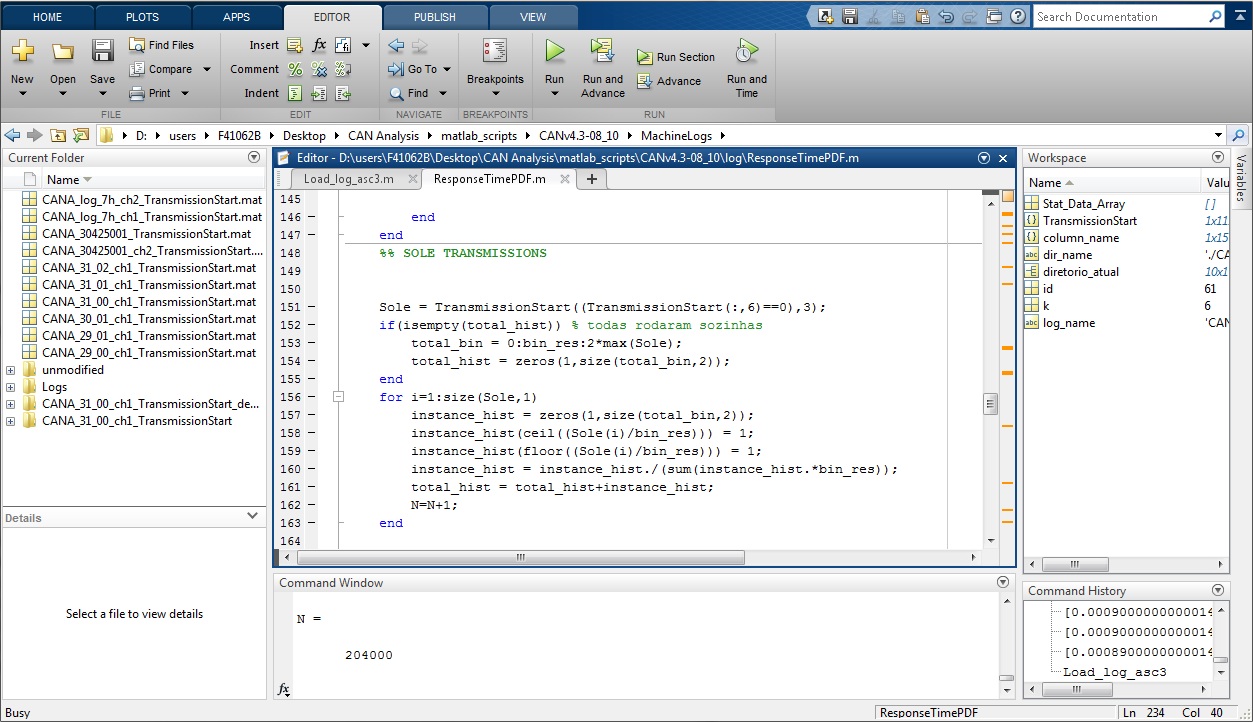


Figura 61 - Interface gráfica do software MATLAB

Fonte: Autoria Própria

Durante o desenvolvimento do método vários programas foram criados para o teste de alguns conceitos, como a implementação das equações e um simulador de protocolo CAN (utilizado também para gerar vários gráficos presentes neste relatório), porém, após o desenvolvimento teórico do método proposto (seção 6.4), devido aos seus benefícios em relação aos anteriores, este foi o escolhido para ser aplicado a um *log* real da colhedora de cana de açúcar da marca Case.

O objetivo do programa implementado é receber um *log* de dados com o tempo final e inicial de execução de diversas instâncias de cada mensagem e fornecer ao usuário as pdfs e cdfs do tempo de resposta de cada uma.

Observa-se que não há a necessidade de informar ao programa características de comportamento das mensagens além dos tempos descritos acima, pois ele consegue estimar automaticamente quais mensagens são periódicas e em que momentos ela deixa de ser transmitida.

## LOGS UTILIZADOS

Os conjuntos de dados com tempos de diversas execuções foram obtidos através do data logger *CANCaseXL* da empresa Vector, o qual foi conectado ao barramento da colhedora de cana durante períodos maiores que 5 horas contíguas, gerando diversos *logs* em diferentes casos de uso.

Existe a necessidade da utilização de *logs* bastante grandes (5 horas ou mais) pois assim aumentamos a confiabilidade da nossa análise devido a abundante quantidade de amostras. Um *log* de 7 horas possui aproximadamente um milhão de execuções de mensagens.



Figura 62 - CANcaseXL - Data logger

Fonte: http://vector.com/portal/medien/cmc/datasheets/CANcaseXL\_log\_DATASHEET\_EN.pdf

Para facilitar a obtenção dos dados pelo MATLAB, houve a necessidade de exportar os *logs* em formato ASCII, e em seguida utilizar um *script* auxiliar para transformar os arquivos ASCII em formatos *mat*.

## RESULTADOS

O software produzido foi executado utilizando uma série de *logs*  diferentes da colhedora de cana de açúcar, a fim de encontrar os gráficos de probabilidade dos tempos de resposta de cada mensagem.

O resultado produzido foi um gráfico de pdf, cdf e erro estimado para cada mensagem pertencente a cada *log*, por isso, em um *log* com 100 tipos de mensagens diferentes, 300 gráficos foram produzidos.

Houve também a implementação de um relatório de estado do barramento, o qual fornecia diversos dados para cada mensagem, como o período da mensagem, o *k-g*  máximo, o número de execuções, o pior tempo de resposta e o limite máximo de tempo de resposta com 95% de probabilidade de ocorrência.

### Mensagens aperiódicas

A pdf abaixo foi obtida através da análise de uma mensagem aperiódica com a trigésima terceira maior prioridade.

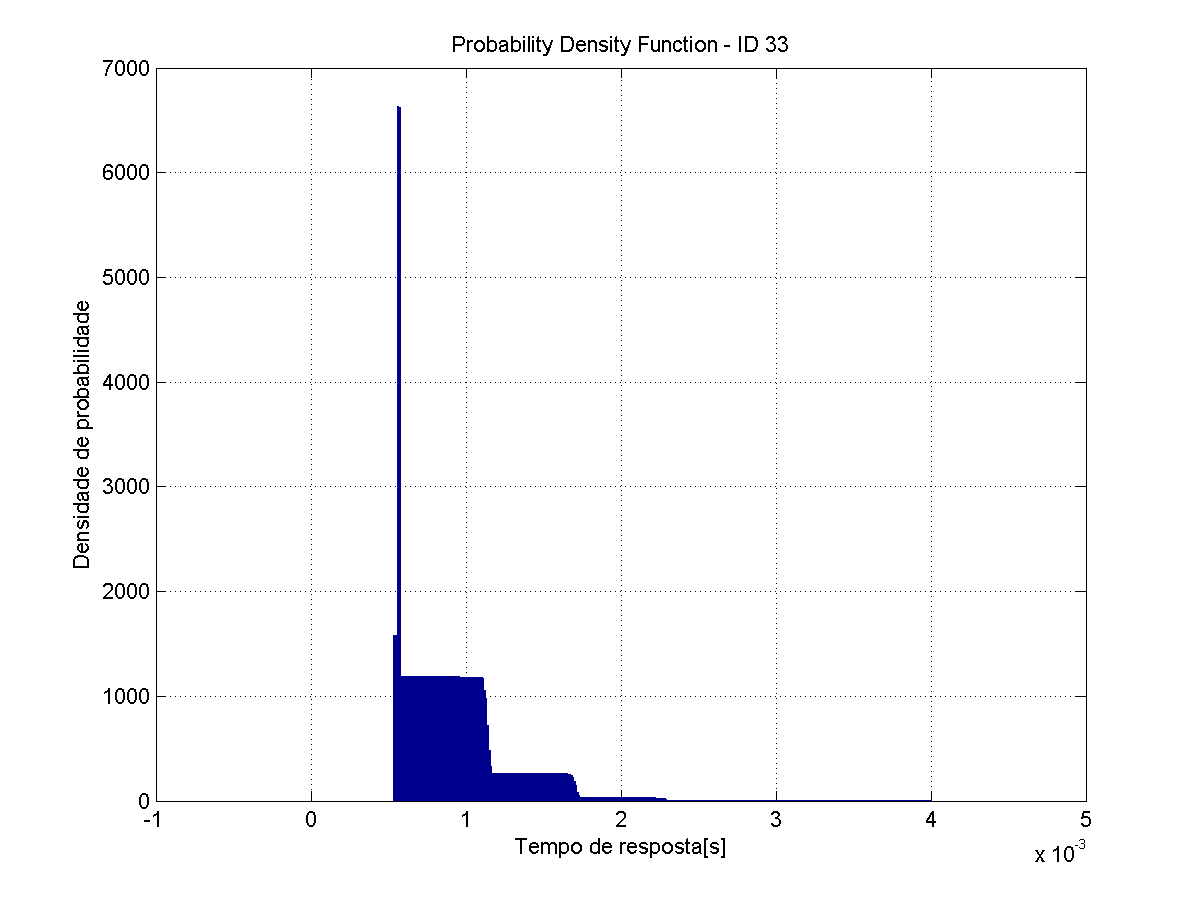


Figura 63 - PDF do tempo de resposta de uma mensagem aperiódica

Fonte: Autoria Própria

Na Figura 63 pode-se observar o efeito da aproximação de tempo ocupado. Essa aproximação resultou em um formato parecido com uma escada, uma vez que existem diversos tempos ocupados que foram modelados como pdfs uniformes, porém, neste caso, observa-se que existiram três tempos ocupados mais predominantes.

A transição entre os degraus presentes é um pouco inclinada e não totalmente retangular pois o tamanho de cada execução presente no barramento é variável, devido aos *stuffing bits*.

Os primeiros valores diferentes de zero no gráfico estão presentes no tempo em que a mensagem não foi atrasada por nenhuma outra e executou com o menor número de *bytes* e de *stuffing bits*, correspondendo ao menor tempo de resposta possível.

Por ser uma mensagem que executou várias vezes sem tempo ocupado (15% do total de execuções), a moda da pdf está situada em um dos seus possíveis valores de tempo de execução.

Através da pdf obtida, calcula-se a cdf abaixo, na qual estão evidenciados o pior tempo de resposta e o limite máximo de tempo de resposta em 95% das execuções.

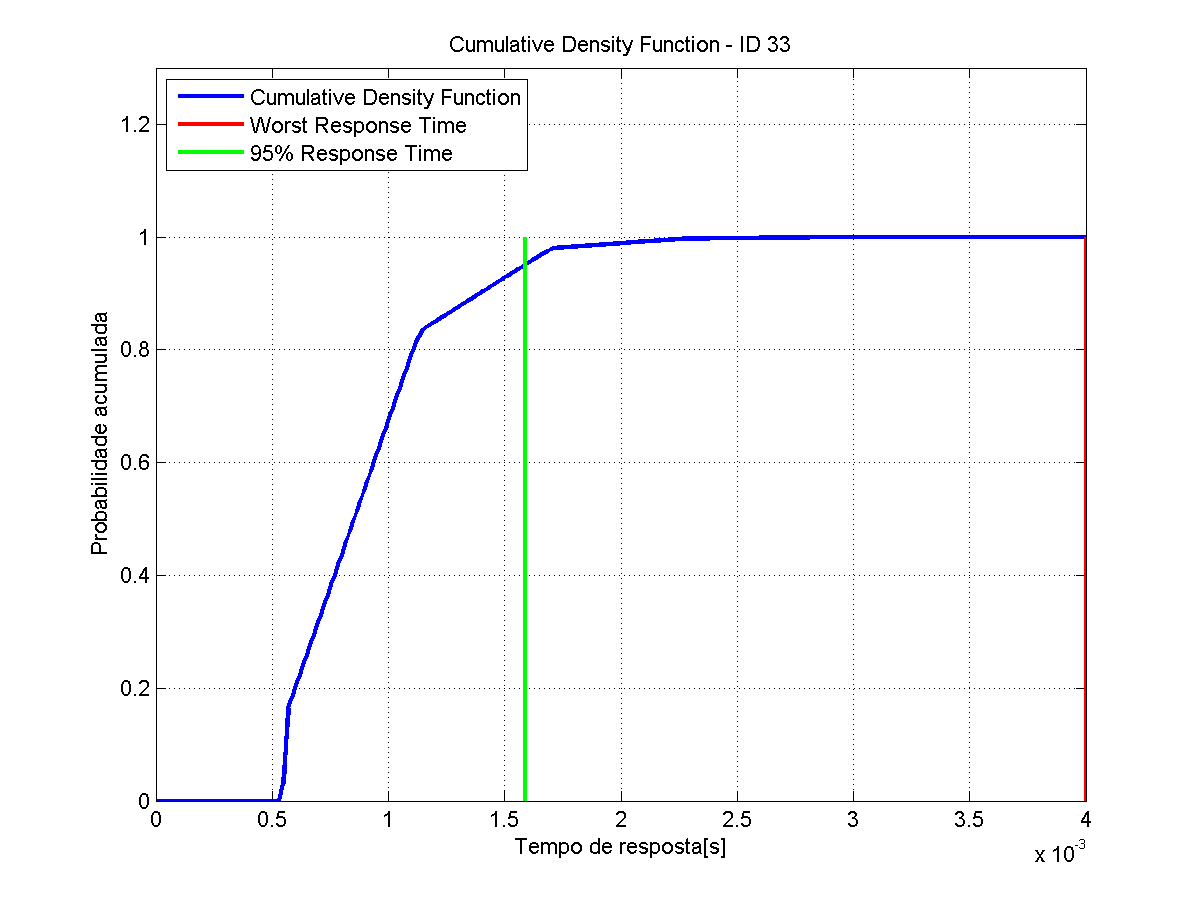


Figura 64 - CDF do tempo de resposta de uma mensagem aperiódica

Fonte: Autoria Própria

A Figura 64 nos fornece um meio de avaliar a relação entre o pior tempo de resposta (em vermelho) e um tempo menos pessimista (em verde), este que é seis vezes menor que o primeiro.

### Mensagens periódicas

A imagem abaixo foi obtida do mesmo *log* que a anterior, analisando uma mensagem periódica com a quadragésima sétima maior prioridade.

Pode-se observar que diferente de anteriormente, apesar do gráfico apresentar alguns degraus, eles não são muito regulares.

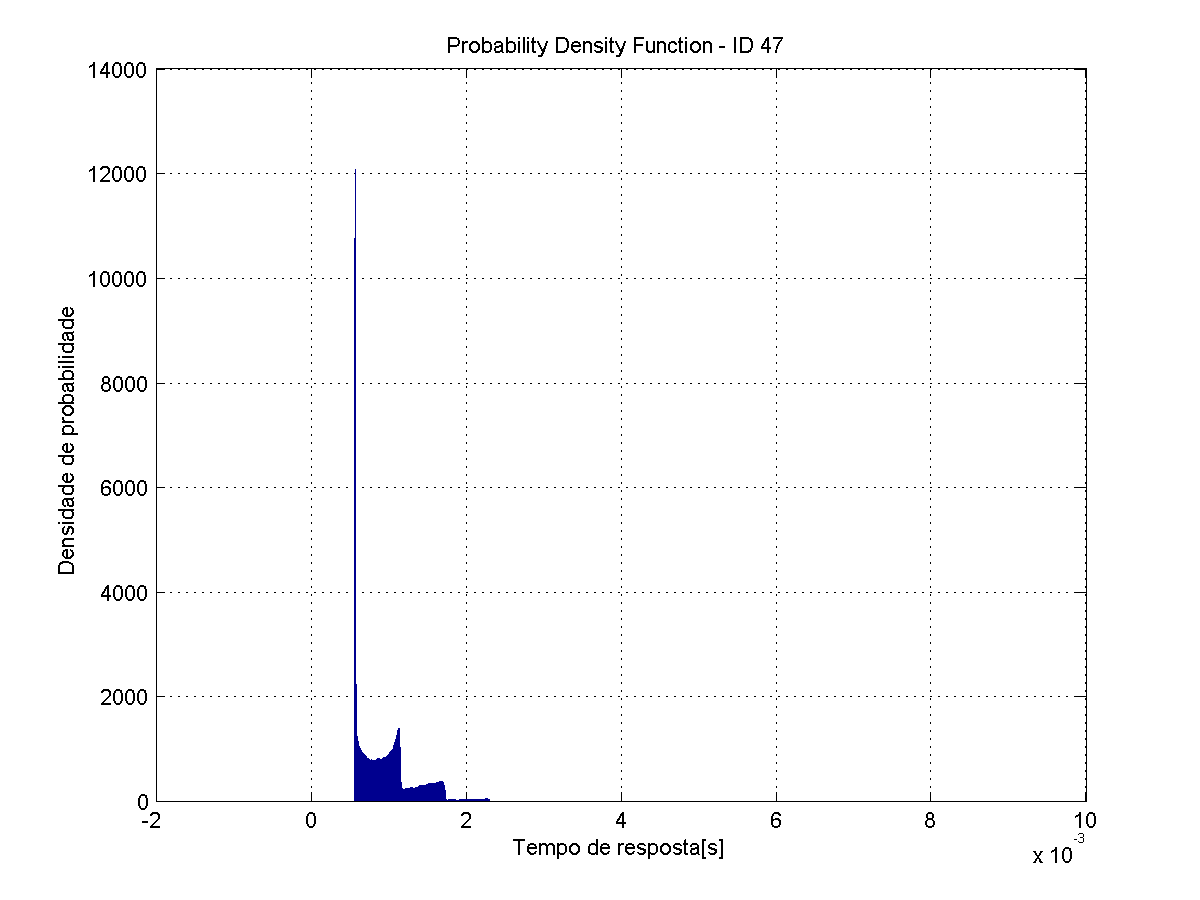


Figura 65 - PDF do tempo de resposta de uma mensagem periódica

Fonte: Autoria Própria

Os degraus ainda existem pois quando *k-g* é maior que o máximo, a projeção de período é feita através de distribuição uniforme, como em uma mensagem aperiódica, porém, esses degraus tem uma forma distinta pois o erro de jitter está sendo utilizado para *k-g* menores que o máximo.

Devido à resolução da Figura 65 de uma mensagem periódica é difícil observar que existem valores diferentes de zero perto do tempo de 9ms, pois foram eventos que ocorreram pouquíssimas vezes quando comparado com o restante do *log*. Apesar disso, são eventos que devem ser considerados em uma análise de tempo de resposta, e a sua baixa incidência resulta em uma baixa probabilidade de ocorrência.

Esses eventos de baixa probabilidade podem ter sido estimados tanto através de um valor alto de jitter ou de tempo ocupado, porém, independente de sua fonte, eles tem uma chance de ocorrer.

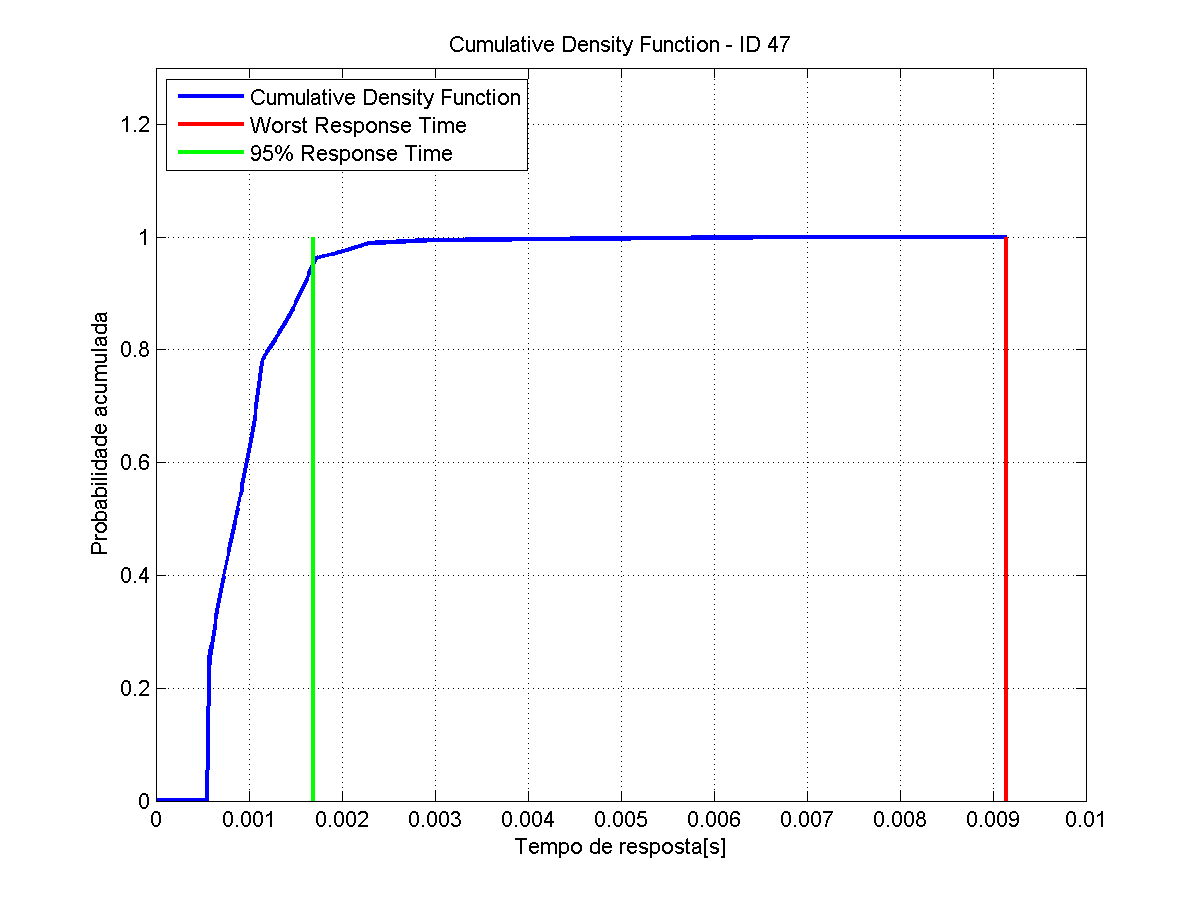


Figura 66 - CDF do tempo de resposta de uma mensagem periódica

Fonte: Autoria Própria

A Figura 66 demonstra a cdf obtida pela pdf anterior, onde pode-se observar a grande diferença entre tempos de resposta para 95% e 100% de ocorrência, sendo o primeiro igual a 1.69ms e aquele 9.14ms, evidenciando mais uma vez como o cálculo de um pior tempo de resposta é pessimista e o design de um barramento através desse parâmetro resulta no seu superdimensionamento.

Na Tabela 11 pode-se observar que o período estimado para essa mensagem foi de 249,88*ms*, que consiste em um valor próximo do teórico (250*ms*), dado que, em conjunto com os níveis de jitter obtidos, certificam que a mensagem está de fato obedecendo uma periodicidade.

### Influência da prioridade de uma mensagem

A Figura 67 mostra a diferença entre tempos de resposta de duas mensagens com prioridades diferentes.

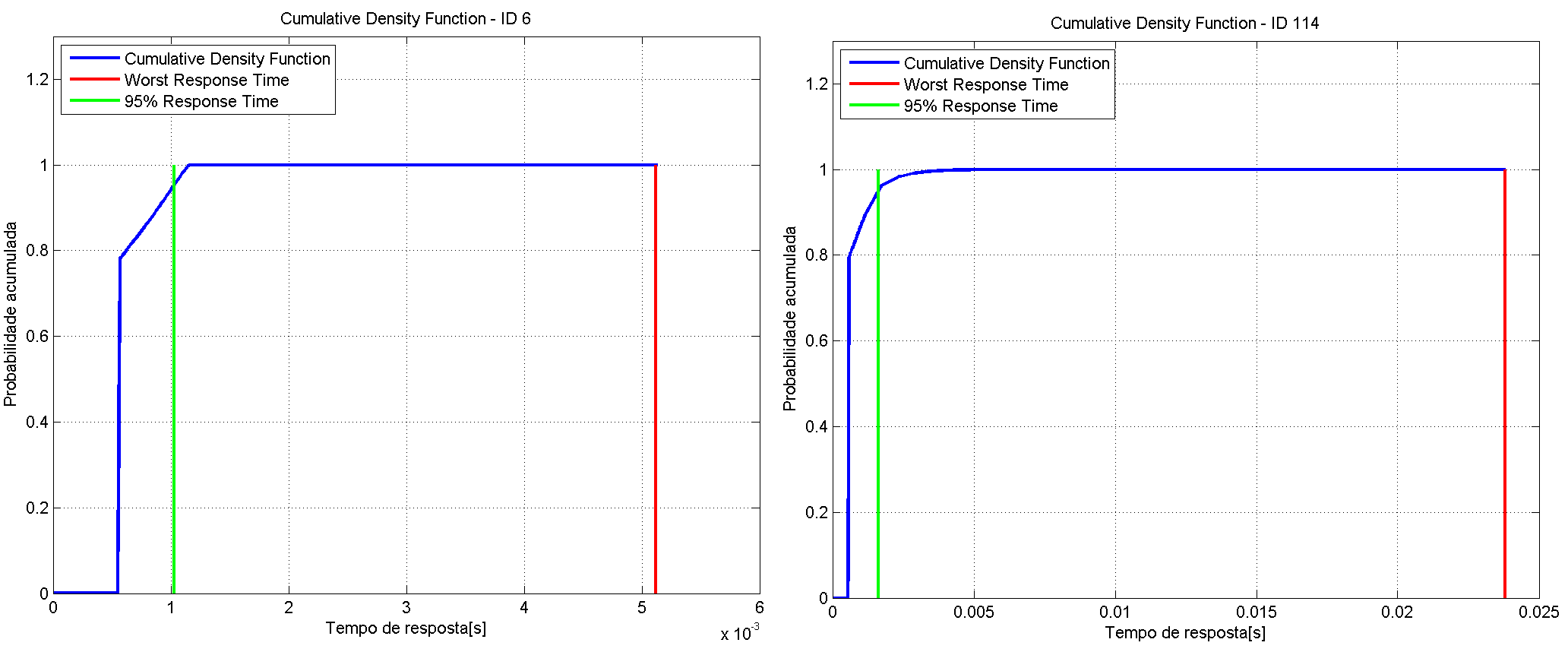


Figura 67 - Diferença de tempos de resposta entre uma mensagem de alta (esquerda) e outra de baixa prioridade (direita)

Fonte: Autoria Própria

Observa-se que o formato do gráfico com maior prioridade é menos suave, pois sofreu menos interferências de mensagens com mais prioridade, além de possuir menores valores de tempo de resposta para as mesmas probabilidades.

# CONSIDERAÇÕES FINAIS

O método proposto cumpriu com o objetivo de estimar as características do barramento e gerar funções de probabilidade através de uma maneira realista e não pessimista, auxiliando na identificação de mensagens que possam causar algum tipo de problema no sistema como um todo devido a um atraso proveniente do barramento CAN.

## COMPARAÇÃO ENTRE MÉTODOS

Para realizar uma comparação entre métodos, houve a tentativa de utilizar os métodos presentes na literatura, porém, estes vieram a falhar no barramento da colhedora de cana de açúcar pois existe uma presença muito grande de mensagens aperiódicas. Esse tipo de mensagem não tem um período fixo, e a aproximação de se utilizar o menor tempo entre invocações sucessivas como um período, resultou em uma utilização de barramento muito maior que 100%, impossibilitando a convergência das equações de Davis et al.

Posteriormente, tentou-se aproximar o tempo entre mensagens aperiódicas à média de diferença de tempo entre invocações sucessivas, o que também não resultou em um sucesso, pois a utilização de barramento foi de 160%.

## CARACTERÍSTICAS DO SOFTWARE DESENVOLVIDO

A performance de execução do programa foi satisfatória devido a sua complexidade, demorando em média 30 minutos por *log* de 7 horas de máquina, com aproximadamente um milhão de execuções.

Uma interface gráfica não foi desenvolvida, porém, isso não diminui a intuitividade de uso do programa pois basta um pequeno conhecimento em MATLAB para que a sua execução seja bastante simples. O *script* é relativamente automatizado, gerando as imagens e o relatório automaticamente, sendo necessário somente transformar o *log* ASCII em .mat e utilizando um código auxiliar.

## INTERPRETAÇÃO DO RELATÓRIO FINAL

O método proposto trata-se de uma ferramenta que fornece as funções de probabilidades para as diversas mensagens de um *log*, porém, para a interpretação e utilização correta dessa ferramenta, faz-se necessário estipular *deadlines* para as diferentes mensagens analisadas.

A estipulação desses *deadlines* não era escopo do trabalho, e para a sua realização devem-se conhecer as funções de cada mensagem e as ECUs envolvidas nessa comunicação, levando em conta os efeitos contrários causados em uma eventual extrapolação do *deadline*.

## TRABALHOS FUTUROS

Vários trabalhos podem ser realizados em função do caminho explorado para a estimativa dos tempos de resposta, tanto em um aprimoramento do método quanto em uma melhor interpretação de seus resultados.

A estipulação de *deadline* para as mensagens é mandatória para a correta interpretação e utilização dos resultados fornecidos pela análise, pois sem ela não existe um parâmetro para decidir um limiar de tempo de resposta aceitável.

A escolha de *deadline* permitirá identificar mensagens que estão em condições mais críticas, o que servirá como base para alterar o modo em que o barramento está disposto, aumentando períodos de mensagens mais prioritárias, diminuindo o número de mensagens, e mudando a ordem de algumas prioridades.

Mesmo sem o conhecimento prévio de *deadline*, estipula-se o período como esse limiar para mensagens periódicas,podendo-se identificar na Tabela 11 ( e através dos gráficos de função de probabilidade) que existem mensagens em uma situação menos confortável que outras, sugerindo mudanças que podem não surtir impacto imediato, mas à medida em que o barramento for sendo mais utilizado, diminuirá a sua chance de problemas, tornando-se uma tarefa altamente recomendável.

Uma interessante espécie de análise seria a integração entre simulador e analisador de *log*, o que possibilitaria “medir” os parâmetros estatísticos do barramento e coloca-los em um simulador, onde podem-se incluir mensagens que antes não existiam para verificar um novo comportamento do barramento, através de um método mais realista, confiável e eficiente.

Tabela 11 - Relatório de saída de uma análise de barramento através do método proposto

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Índice | Período [ms] | Total de execuções | Porcentagem de execuções sem tempo ocupado | Execuções estimadas utilizando uma distribuição uniforme | Variância de Et [ms²] | Variância de Eq [ms²] | k-g máximo | Pior tempo de resposta [ms] | Limite máximo para tempo de resposta em 95% das execuções [s] |
| 1 | -- | 4 | 50,00 | 2 | 0,00 | 0,00 | 0 | 1,070 | 0,960 |
| 2 | -- | 2 | 50,00 | 1 | 0,00 | 0,00 | 0 | 0,960 | 0,920 |
| 3 | -- | 2 | 50,00 | 1 | 0,00 | 0,00 | 0 | 1,030 | 0,980 |
| 4 | -- | 2 | 0,00 | 2 | 0,00 | 0,00 | 0 | 1,050 | 1,020 |
| 5 | -- | 30 | 13,33 | 26 | 0,00 | 0,00 | 0 | 1,150 | 1,090 |
| 6 | 20,00 | 951754 | 78,16 | 5655 | 0,01 | 1,27 | 30 | 5,120 | 1,030 |
| 7 | 20,00 | 951752 | 73,23 | 7899 | 0,01 | 0,68 | 30 | 25,280 | 1,050 |
| 8 | 49,97 | 321788 | 23,88 | 16774 | 0,00 | 0,82 | 30 | 26,010 | 1,140 |
| 9 | 19,99 | 804456 | 47,46 | 13960 | 0,00 | 0,39 | 30 | 3,250 | 1,350 |
| 10 | -- | 294560 | 51,48 | 142907 | 306,36 | 392,75 | 0 | 3,470 | 1,120 |
| 11 | 50,00 | 380699 | 70,56 | 4023 | 0,00 | 2,95 | 30 | 4,050 | 1,110 |
| 12 | -- | 243434 | 41,93 | 141359 | 118,86 | 223,33 | 0 | 4,050 | 1,180 |
| 13 | -- | 64 | 42,19 | 37 | 0,00 | 0,00 | 0 | 2,290 | 1,220 |
| 14 | -- | 64 | 42,19 | 37 | 0,00 | 0,00 | 0 | 2,270 | 1,430 |
| 15 | -- | 26 | 26,92 | 19 | 0,00 | 0,00 | 0 | 2,300 | 1,420 |
| 16 | -- | 22 | 13,64 | 19 | 0,00 | 0,00 | 0 | 1,700 | 1,140 |
| 17 | -- | 22 | 22,73 | 17 | 0,00 | 0,00 | 0 | 2,290 | 1,340 |
| 18 | -- | 22 | 18,18 | 18 | 0,00 | 0,00 | 0 | 2,280 | 1,180 |
| 19 | -- | 22 | 54,55 | 10 | 0,00 | 0,00 | 0 | 1,150 | 1,080 |
| 20 | -- | 16 | 25,00 | 12 | 0,00 | 0,00 | 0 | 1,550 | 1,240 |
| 21 | -- | 4 | 75,00 | 1 | 0,00 | 0,00 | 0 | 0,970 | 0,860 |
| 22 | -- | 32 | 50,00 | 16 | 0,00 | 0,00 | 0 | 1,710 | 1,340 |
| 23 | -- | 2 | 50,00 | 1 | 0,00 | 0,00 | 0 | 1,130 | 1,080 |
| 24 | 1000,05 | 19032 | 78,97 | 42 | 5,60 | 177,50 | 30 | 154,400 | 1,070 |
| 25 | -- | 22 | 27,27 | 16 | 0,00 | 0,00 | 0 | 1,690 | 1,380 |
| 26 | -- | 22 | 59,09 | 9 | 0,00 | 0,00 | 0 | 1,690 | 1,080 |
| 27 | -- | 22 | 22,73 | 17 | 0,00 | 0,00 | 0 | 3,430 | 1,650 |
| 28 | -- | 22 | 59,09 | 9 | 0,00 | 0,00 | 0 | 3,390 | 2,060 |
| 29 | 999,85 | 19039 | 81,50 | 0 | 1,32 | 53,74 | 30 | 796,510 | 0,840 |
| 30 | 199,99 | 95189 | 80,11 | 86 | 0,01 | 0,68 | 30 | 104,200 | 0,860 |
| 31 | -- | 4 | 50,00 | 2 | 0,00 | 0,00 | 0 | 0,970 | 0,910 |
| 32 | -- | 12 | 33,33 | 8 | 0,00 | 0,00 | 0 | 1,520 | 0,950 |
| 33 | -- | 12891 | 15,85 | 10848 | 0,00 | 0,00 | 0 | 4,000 | 1,590 |
| 34 | 4976,44 | 3223 | 7,79 | 2824 | 0,00 | 435217,05 | 30 | 4336,720 | 1,690 |
| 35 | -- | 23 | 52,17 | 11 | 0,00 | 0,00 | 0 | 1,150 | 1,080 |
| 36 | -- | 9 | 0,00 | 9 | 0,00 | 0,00 | 0 | 1,740 | 1,210 |
| 37 | -- | 20 | 25,00 | 15 | 0,00 | 0,00 | 0 | 2,680 | 1,530 |
| 38 | -- | 23 | 65,22 | 8 | 0,00 | 0,00 | 0 | 2,650 | 1,360 |
| 39 | -- | 40 | 32,50 | 27 | 0,00 | 0,00 | 0 | 3,430 | 1,420 |
| 40 | -- | 15 | 40,00 | 9 | 0,00 | 0,00 | 0 | 1,720 | 1,120 |
| 41 | -- | 20 | 25,00 | 15 | 0,00 | 0,00 | 0 | 2,270 | 1,700 |
| 42 | -- | 24 | 54,17 | 11 | 0,00 | 0,00 | 0 | 1,730 | 1,140 |
| 43 | -- | 15 | 46,67 | 8 | 0,00 | 0,00 | 0 | 1,150 | 1,090 |
| 44 | -- | 15 | 26,67 | 11 | 0,00 | 0,00 | 0 | 1,690 | 1,260 |
| 45 | 999,76 | 16090 | 5,08 | 13794 | 0,00 | 87,95 | 30 | 150,100 | 2,210 |
| 46 | 999,46 | 16088 | 6,58 | 12493 | 0,00 | 0,21 | 30 | 4,610 | 1,680 |
| 47 | 249,88 | 64360 | 23,90 | 590 | 0,03 | 0,77 | 30 | 9,140 | 1,690 |
| 48 | 999,52 | 16089 | 42,86 | 902 | 0,41 | 5,77 | 28 | 104,570 | 1,140 |
| 49 | -- | 16091 | 7,31 | 14915 | 5,91 | 0,16 | 0 | 4,620 | 2,180 |
| 50 | 499,90 | 32180 | 7,78 | 24673 | 1,03 | 14,80 | 28 | 149,540 | 2,180 |
| 51 | 99,95 | 160902 | 6,90 | 127731 | 0,00 | 0,05 | 30 | 5,930 | 2,490 |
| 52 | 999,51 | 16091 | 43,33 | 3821 | 0,17 | 0,37 | 4 | 58,870 | 1,150 |
| 53 | 499,72 | 32181 | 7,71 | 22195 | 0,09 | 24,99 | 30 | 154,210 | 2,740 |
| 54 | 999,41 | 16092 | 42,93 | 856 | 3,44 | 381,41 | 30 | 901,370 | 1,150 |
| 55 | 1000,00 | 19032 | 77,36 | 0 | 0,85 | 39,49 | 30 | 3,990 | 1,080 |
| 56 | -- | 949984 | 48,39 | 490264 | 0,39 | 0,01 | 0 | 5,770 | 1,170 |
| 57 | 20,00 | 951753 | 77,41 | 5205 | 0,00 | 0,97 | 30 | 18,770 | 1,100 |
| 58 | 100,12 | 190036 | 42,74 | 5121 | 0,00 | 0,41 | 30 | 146,070 | 1,470 |
| 59 | 20,03 | 949933 | 81,17 | 3249 | 0,01 | 4,41 | 30 | 9,440 | 1,030 |
| 60 | 60,02 | 267749 | 45,07 | 7412 | 0,07 | 4,11 | 30 | 57,780 | 1,540 |
| 61 | 49,97 | 321780 | 23,80 | 18807 | 0,00 | 0,79 | 30 | 12,480 | 1,960 |
| 62 | 20,01 | 803253 | 42,90 | 143133 | 0,16 | 1,09 | 13 | 26,520 | 1,830 |
| 63 | 60,02 | 267749 | 37,26 | 77913 | 0,00 | 0,17 | 30 | 9,240 | 2,060 |
| 64 | 500,13 | 32129 | 38,38 | 112 | 0,00 | 1,46 | 30 | 10,760 | 2,190 |
| 65 | 199,98 | 95180 | 43,63 | 11821 | 0,00 | 0,01 | 30 | 8,050 | 1,680 |
| 66 | 199,99 | 95177 | 72,28 | 2053 | 0,01 | 0,34 | 30 | 126,940 | 1,330 |
| 67 | 199,99 | 95180 | 41,41 | 16074 | 0,02 | 0,70 | 30 | 153,740 | 1,730 |
| 68 | -- | 1816 | 33,48 | 1208 | 0,00 | 0,00 | 0 | 6,320 | 1,690 |
| 69 | -- | 10 | 100,00 | 0 | 0,00 | 0,00 | 0 | 0,580 | 0,580 |
| 70 | -- | 92 | 38,04 | 57 | 0,00 | 0,00 | 0 | 4,030 | 1,750 |
| 71 | 199,99 | 95179 | 40,48 | 11698 | 0,03 | 1,37 | 30 | 135,220 | 1,670 |
| 72 | 20,03 | 949926 | 76,66 | 12350 | 0,01 | 1,98 | 30 | 11,480 | 1,240 |
| 73 | 999,90 | 19037 | 27,27 | 8664 | 0,00 | 0,04 | 30 | 9,210 | 2,290 |
| 74 | 50,00 | 380719 | 44,69 | 32046 | 0,00 | 0,02 | 30 | 23,790 | 1,750 |
| 75 | -- | 190360 | 43,67 | 107221 | 0,00 | 0,00 | 0 | 16,140 | 1,780 |
| 76 | 199,98 | 95180 | 42,29 | 13923 | 0,00 | 0,00 | 30 | 18,470 | 1,740 |
| 77 | -- | 49951 | 41,54 | 29203 | 0,00 | 0,00 | 0 | 4,610 | 1,150 |
| 78 | -- | 50438 | 71,56 | 14345 | 0,00 | 0,00 | 0 | 9,790 | 1,090 |
| 79 | -- | 914 | 21,23 | 720 | 0,00 | 0,00 | 0 | 21,400 | 2,840 |
| 80 | -- | 50438 | 30,69 | 34958 | 0,00 | 0,00 | 0 | 16,730 | 1,530 |
| 81 | 199,99 | 95175 | 73,74 | 569 | 0,06 | 1,34 | 30 | 270,870 | 1,520 |
| 82 | 50,00 | 380699 | 72,04 | 2650 | 0,00 | 3,23 | 30 | 11,550 | 1,770 |
| 83 | 199,99 | 95174 | 78,30 | 676 | 0,08 | 1,24 | 30 | 103,810 | 1,310 |
| 84 | 200,00 | 95172 | 72,29 | 680 | 0,22 | 4,92 | 30 | 60,310 | 1,660 |
| 85 | 50,00 | 380701 | 71,81 | 4995 | 0,00 | 2,73 | 30 | 14,350 | 1,770 |
| 86 | 199,99 | 95173 | 79,32 | 234 | 0,05 | 1,72 | 30 | 176,340 | 1,350 |
| 87 | 199,99 | 95174 | 71,37 | 1436 | 0,14 | 3,31 | 30 | 203,070 | 1,610 |
| 88 | 199,99 | 95173 | 77,03 | 660 | 0,35 | 6,74 | 30 | 157,440 | 1,390 |
| 89 | 199,99 | 95175 | 76,97 | 446 | 1,20 | 51,33 | 30 | 19,730 | 1,440 |
| 90 | -- | 190352 | 70,87 | 55455 | 0,00 | 0,00 | 0 | 19,620 | 1,990 |
| 91 | -- | 285523 | 74,22 | 73621 | 0,00 | 0,00 | 0 | 21,980 | 1,850 |
| 92 | 999,92 | 19032 | 79,62 | 1700 | 2,54 | 3,67 | 2 | 8,660 | 1,600 |
| 93 | 199,99 | 95175 | 70,36 | 1807 | 2,06 | 105,28 | 30 | 8,140 | 1,660 |
| 94 | 50,00 | 380718 | 44,02 | 43003 | 0,00 | 0,01 | 30 | 9,760 | 2,140 |
| 95 | -- | 190360 | 43,30 | 107927 | 0,00 | 0,00 | 0 | 22,580 | 2,290 |
| 96 | 199,99 | 95180 | 43,01 | 14598 | 0,02 | 0,69 | 30 | 154,150 | 2,140 |
| 97 | 10,00 | 1903552 | 76,97 | 3 | 0,00 | 2,95 | 30 | 13,860 | 1,890 |
| 98 | 10,00 | 1903545 | 79,98 | 6 | 0,00 | 3,16 | 30 | 15,010 | 1,660 |
| 99 | -- | 122 | 15,57 | 103 | 0,00 | 0,00 | 0 | 6,330 | 3,280 |
| 100 | -- | 124 | 46,77 | 66 | 0,00 | 0,00 | 0 | 9,230 | 3,440 |
| 101 | -- | 6 | 66,67 | 2 | 0,00 | 0,00 | 0 | 4,030 | 3,250 |
| 102 | -- | 135 | 22,96 | 104 | 0,00 | 0,00 | 0 | 13,260 | 4,360 |
| 103 | -- | 30 | 3,33 | 29 | 0,00 | 0,00 | 0 | 16,750 | 6,790 |
| 104 | -- | 1336 | 44,16 | 746 | 0,00 | 0,00 | 0 | 9,770 | 2,680 |
| 105 | -- | 170 | 63,53 | 62 | 0,00 | 0,10 | 0 | 9,810 | 2,510 |
| 106 | -- | 79 | 55,70 | 35 | 0,00 | 0,00 | 0 | 11,920 | 2,830 |
| 107 | -- | 40 | 25,00 | 30 | 0,00 | 0,00 | 0 | 8,540 | 4,960 |
| 108 | -- | 67 | 29,85 | 47 | 0,00 | 0,00 | 0 | 6,910 | 3,400 |
| 109 | -- | 2 | 50,00 | 1 | 0,00 | 0,00 | 0 | 2,910 | 2,680 |
| 110 | 1017,16 | 15778 | 41,03 | 90 | 71,73 | 474,30 | 13 | 446,760 | 50,710 |
| 111 | 1000,42 | 16063 | 44,75 | 26 | 0,00 | 0,01 | 30 | 15,250 | 2,250 |
| 112 | 1000,38 | 16063 | 41,72 | 61 | 0,00 | 0,08 | 30 | 17,830 | 3,490 |
| 113 | 1000,31 | 16075 | 42,72 | 9206 | 25,45 | 19,29 | 1 | 279,290 | 3,840 |
| 114 | 20,00 | 951779 | 79,29 | 5 | 0,00 | 2,43 | 30 | 23,800 | 1,630 |

Obs: mensagens com o período -- correspondem a aperiódicas

# Referências

JOHANSSON, KARL H.; Torngren, Martin; Nielsen, Lars. **Vehicle Applications of Controller Area Network**. Disponível em: <https://people.kth.se/~kallej/papers/can\_necs\_handbook05.pdf.>. Acesso em: 20 set. 2015.

CAN in Automation, CiA. **CAN data link layers**. Am Weichselgarten 26, D-91058 Erlangen, 2002. Disponível em <http://www.can-cia.org/can-knowledge/can/can-data-link-layers/>. Acesso em: 20 set. 2015.

CNHi. **New Holland T7**. 2014. 9p. Disponível em: <http://agriculture.newholland.com/br/pt/Products/Tractors/T7/Documents/t7\_sps.pdf.>. Acesso em: 21 set. 2015.

TINDEL, A. Ken. **An Extendible Approach for Analysing Fixed Priority Hard Real-Time Tasks** . - Universidade de York : Dezembro de 1992.

DAVIS, Robert I. [et al.]**Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised** .Real-Time Systems. - 2007. - Vol. 35. - pp. 239-272.

JOSEPH, M ; PANDYA P. **Finding Response Times in Real-Time System**. BCS Computer Journal. - Outubro de 1986. - Vol. 29. - pp. 390-395.

LEHOCZKY, John P. **Fixed priority scheduling of periodic task sets with arbitrary deadlines** .  Proceedings 11th IEEE Real-Time Systems Symposium. - 5-7 de Dezembro de 1990. - pp. 201-209.

HARBOUR MG; KLEIN MH e JP LEHOCZKY. **Fixed priority scheduling of periodic tasks with varying execution priority**. Proceedings 12th IEEE real time systems symposium. - [s.l.] : IEEE Computer Society Press, 1991. - pp. 116-128.

TINDEL, A. ; BURNS, Alan. **Guaranteeing message latencies on Controller Area Network (CAN)**. Proceedings of 1st international CAN conference. - 1994. - pp. 1-11.

LIU C. L. ; LAYLAND J.W. **Scheduling algorithms for multiprogramming in hard-real-time environment**. Journal of the ACM. - 1973. - Vol. 20. - pp. 46-61.