

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

Daniel Rossi Korol

Monitoramento sem fio da vibração torsional de powertrain automotivo

Joinville

2015.

Daniel Rossi Korol

Monitoramento sem fio da vibração torsional de powertrain automotivo

Trabalho de Conclusão de Curso submetido ao Curso de Engenharia Mecatrônica da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia Mecatrônica.

Orientador: Dr. Anderson Wedderhoff Spengler
Co-orientador: Dr. Giovanni Gracioli.

Joinville

2015.

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Korol, Daniel Rossi
Monitoramento sem fio da vibração torsional de
powertrain automotivo / Daniel Rossi Korol ; orientador,
Anderson Wedderhoff Spengler ; coorientador, Giovanni
Gracioli. - Joinville, SC, 2015.
61 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Joinville.
Graduação em Engenharia Mecatrônica.

Inclui referências

1. Engenharia Mecatrônica. 2. Instrumentação. 3. Vibração
torsional. 4. Powertrain. 5. Rede de sensores sem fio. I.
Spengler, Anderson Wedderhoff. II. Gracioli, Giovanni. III.
Universidade Federal de Santa Catarina. Graduação em
Engenharia Mecatrônica. IV. Título.

Daniel Rossi Korol

Monitoramento sem fio da vibração torsional de powertrain automotivo

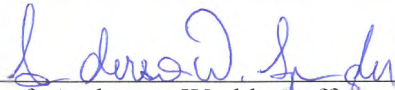
Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Engenheiro Mecatrônico, e aprovado em sua forma final pelo Programa de Graduação.

Joinville, 25 de novembro de 2015.



Prof. Xisto Lucas Travassos Junior, Dr.
Coordenador do Curso

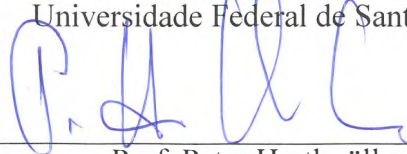
Banca Examinadora:




Prof. Anderson Wedderhoff Spengler, Dr.
Orientador
Universidade Federal de Santa Catarina



Prof. Sérgio Junichi Idehara, Dr.
Universidade Federal de Santa Catarina



Prof. Peter Hartlmüller, Dr.
Technische Hochschule Ingolstadt



Petroneo Pereira, Msc.
BMW do Brasil

A Marcos, Mara, André e Laura.

AGRADECIMENTOS

A Deus por permitir que essa etapa fosse cumprida.

Aos meus pais Marcos Wolfgang Braziellas Korol e Mara Silvia Rossi Korol, ao meu irmão André Rossi Korol, à minha namorada Laura Magalhães Vasconcellos e a toda minha família pelo apoio oferecido durante minha graduação.

Aos professores Anderson Wedderhoff Spengler, Giovani Gracioli e Sérgio Idehara por terem me orientado e suportado durante este trabalho.

Aos meus colegas de trabalho Petroneo Pereira e Marcelo Henrique Quiquio pelo suporte que deram na validação deste trabalho.

Ao pessoal da engenharia experimental da empresa ZF do Brasil, em especial a Wilson Rodrigo de Souza e Eduardo da Silva, pelos testes realizados em conjunto.

Aos meus colegas de faculdade pela amizade e pela convivência durante esta jornada.

RESUMO

A instalação de sensores para a medição da vibração torsional em diversos pontos do *powertrain* se faz necessária em testes veiculares, o que consome tempo considerável da engenharia experimental. Durante essa instrumentação, alguns locais onde os sensores devem ser instalados podem ser de difícil acesso e, em alguns casos, existe a necessidade de se fazer furos na estrutura do veículo para permitir a passagem de cabos dos sensores para o sistema de aquisição de dados, que usualmente encontra-se no *cockpit* do veículo. Essa passagem de cabos está sujeita a inúmeras falhas, como a ruptura acidental, a interferência eletromagnética, o mal funcionamento com o tempo, entre outros. Nesse sentido, a eliminação da ligação física entre os sensores e o sistema de aquisição de dados constitui em uma solução para os problemas apontados anteriormente. Portanto, propõe-se aqui uma comunicação sem fio a fim de simplificar a instrumentação para o monitoramento da vibração torsional de *powertrain* automotivo, através de módulos de comunicação EPOSMoteII.

Palavras-chave: Instrumentação, Vibração torsional, *Powertrain*, Rede de sensores sem fio, Ruído de *rattle*.

ABSTRACT

In vehicle testing, there is the necessity of placing sensors to measure the torsional vibration in several positions in the powertrain and this takes substantial time of experimental engineering. During this instrumentation, it may be difficult to access some places where the sensors need to be placed and it might be even required to make some borings on the vehicle's body for guiding cables from the sensors to the data acquisition system, which is usually positioned in the cockpit. The cable route may lead to several failures, as accidental disruption, electromagnetic interference, malfunction after some time, among others. Thus, eliminating the physical connection between the sensors and the data acquisition system is a worthy solution for the problems described above. Therefore, this work proposes a wireless instrumentation using EPOSMoteII communication boards in order to simplify the vehicle's powertrain torsional vibration measurement.

Keywords: *Instrumentation, Torsional vibration, Powertrain, Wireless sensor network, Rattle noise.*

LISTA DE FIGURAS

Figura 1 – Transmissão instrumentada em uma bancada de testes.	4
Figura 2 – Experimento para a medição da vibração em um eixo.	5
Figura 3 – Rede de sensores sem fio aplicada para o controle ativo da vibração da estrutura de um veículo.	6
Figura 4 – Diagrama esquemático de um sensor magnético do tipo <i>pick-up</i>	8
Figura 5 – Restrições para a instalação de um sensor magnético do tipo <i>pick-up</i>	9
Figura 6 – Sinal gerado pelo sensor magnético do tipo <i>pick-up</i> em teste realizado em torno.	10
Figura 7 – Parte do diagrama esquemático da placa de condicionamento de sinal com ênfase no amplificador operacional e no comparador.	11
Figura 8 – Parte do diagrama esquemático da placa de condicionamento de sinal com ênfase nos reguladores de tensão.	12
Figura 9 – Teste realizado para avaliar o funcionamento da placa de condicionamento de sinal.	13
Figura 10 – Comparação entre as versões EPOSMoteI (a) e EPOSMoteII (b).	14
Figura 11 – Trecho do programa 1 com as funções do <i>timer</i> e do alarme.	16
Figura 12 – Programa 1 – Otimização do parâmetro <i>comp1</i>	17
Figura 13 – Programa 1 – Otimização do parâmetro <i>ALARM_PERIOD</i>	18
Figura 14 – Trecho do programa 2 com as funções do <i>timer</i> e do alarme.	19
Figura 15 – Programa 2 – Otimização do parâmetro <i>ALARM_PERIOD</i>	19
Figura 16 – Trecho do programa 3 com a função do <i>timer</i>	20
Figura 17 – Programa 3 – Otimização do parâmetro <i>comp1</i>	21
Figura 18 – Comparação dos erros relativos entre os 3 programas.	22
Figura 19 – Programa 3 – Otimização do parâmetro <i>comp1</i> para uma faixa de frequências de 1500Hz a 15500Hz.	23
Figura 20 – Programa 3 – Otimização do parâmetro <i>comp1</i> para uma faixa de frequências de 400Hz a 4133Hz.	25
Figura 21 – Trecho do programa 3 otimizado para ler médias através de número fixo de amostras.	26
Figura 22 – Otimização do programa 3 – Médias através de número fixo de amostras.	27
Figura 23 – Trecho do programa 3 modificado para ler médias por tempo.	28
Figura 24 – Otimização do programa 3 – Médias através de período de alarme.	29
Figura 25 – Trecho do programa que define a estrutura da mensagem enviada.	30

Figura 26 – Trecho do programa que contém a função para envio de mensagens.....	31
Figura 27 – Trecho do programa para recebimento de mensagens.	31
Figura 28 – Otimização do parâmetro N_SAMPLES para a placa 1.	32
Figura 29 – Otimização do parâmetro N_SAMPLES para a placa 2.	33
Figura 30 – Trecho do programa final da placa 1 que mostra a estrutura para mudança de faixa.	35
Figura 31 – Teste do programa final para a placa 1.	36
Figura 32 – Teste do programa final para a placa 2.	37
Figura 33 – Passagem de cabos necessária para conectar o Rotec aos sensores.	39
Figura 34 – Sistema próprio posicionado sob o capô do veículo; #1 – Bateria de 9V; #2 – Placa de condicionamento de sinal mais EPOSMoteII; #3 – Conectores BNC para o sensor.	40
Figura 35 – Resultado do teste em marcha lenta com o motor quente realizado na empresa ZF do Brasil.....	41
Figura 36 – Resultado do teste de tração em 2ª marcha realizado na empresa ZF do Brasil. ...	42
Figura 37 – Resultado do teste de tração em 3ª marcha realizado na empresa ZF do Brasil. ...	43
Figura 38 – Instrumentação com o sistema próprio; #1 – Sensor magnético; #2 – Placa de comunicação EPOSMoteII mais placa de condicionamento; #3 – Bateria de 9V.....	45
Figura 39 – Instrumentação com o sistema da fabricante Müller-BBM; #1 – PAK MKII; #2 – Conversor de sinais.....	46
Figura 40 – Resultado do teste em marcha lenta realizado na empresa BMW do Brasil.....	47
Figura 41 – Resultado do teste de tração em 2ª marcha realizado na empresa BMW do Brasil.	48
Figura 42 – Resultado do teste com troca de marchas da 1ª até a 8ª marcha realizado na empresa BMW do Brasil.	49

LISTA DE TABELAS

Tabela 1 – Faixas de operação da placa 1 e seus respectivos valores de comp1.....	24
Tabela 2 – Faixas de operação da placa 2 e seus respectivos valores de comp1.....	25
Tabela 3 – Definição do parâmetro N_SAMPLES para cada faixa de operação da placa 1....	33
Tabela 4 – Definição do parâmetro N_SAMPLES para cada faixa de operação da placa 2....	34
Tabela 5 – Comparativo entre o Rotec e o sistema próprio.....	44
Tabela 6 – Comparativo entre o Müller-BBM alimentado pelo conversor de sinais e o sistema próprio.	50

LISTA DE SIGLAS

ADC	Analog to Digital Converter
ADESD	Application-Driven Embedded System Design
CAN	Controller Area Network
EPOS	Embedded Parallel Operating System
LISHA	Laboratório de Integração Software/Hardware
MEMS	Micro Electro Mechanical System
NIC	Network Interface Controller
OBD	On Board Diagnostic
RAM	Random Access Memory
RAS	Rotation Analysis System
RF	Rádio Frequência
ROM	Read Only Memory
SPI	Serial Peripheral Interface
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network

SUMÁRIO

1. INTRODUÇÃO	1
2. REFERENCIAL TEÓRICO	3
3. METODOLOGIA	7
3.1. Condições de teste	7
3.2. Sensores magnéticos do tipo <i>pick-up</i>	7
3.3. Placas de condicionamento de sinal.....	10
3.4. Módulos de comunicação EPOSMoteII	13
3.5. Projeto e desenvolvimento do <i>software</i>	15
3.5.1. Leitura do sensor magnético	15
3.5.1.1. Programa 1	15
3.5.1.2. Programa 2.....	18
3.5.1.3. Programa 3.....	20
3.5.1.4. Otimização do programa 3	23
3.5.2. Envio de mensagens	30
3.5.3. Programa final	32
4. RESULTADOS	38
4.1. Testes realizados na empresa ZF do Brasil.....	38
4.2. Testes realizados na empresa BMW do Brasil	44
5. CONCLUSÃO	51
REFERÊNCIAS	53
APÊNDICE A	55
APÊNDICE B.....	56
APÊNDICE C.....	59

1. INTRODUÇÃO

A tendência mundial de se aumentar a potência dos motores à combustão para melhoria do desempenho veicular, em adição à busca pela redução de peso, tem impactado na elevação dos níveis de vibração e ruído nos veículos. Por outro lado, a demanda crescente dos clientes por veículos mais silenciosos resultou na necessidade de se entender melhor o comportamento dinâmico do *powertrain* e de se reduzir sua vibração torsional para níveis aceitáveis (COUDERC et al., 1998).

De fato, se os clientes pagam por um sistema de áudio ou entretenimento acima do mercado, certamente querem apreciar o que compraram. Nesse sentido, uma das principais fontes de insatisfação dos clientes e, conseqüentemente, um requisito importante para a engenharia das montadoras é o ruído gerado na transmissão. Esse ruído é excitado pela vibração torsional que se propaga do motor à combustão para todo o *powertrain* veicular (COUDERC et al., 1998).

A avaliação do nível de vibração e a definição dos parâmetros construtivos das embreagens automotivas para controlar ou minimizar esse efeito no *powertrain* é feito através de testes padronizados em veículos durante a fase de desenvolvimento do produto. Em muitos testes veiculares, requer-se a passagem de sensores para a medição da vibração em diversos pontos do *powertrain*. A passagem de cabos para a realização desses testes tem se mostrado problemática, logo, o emprego de tecnologia de comunicação sem fio apresenta grande potencial para reduzir o tempo da engenharia experimental. Além disso, a necessidade de se fazer furos na estrutura do veículo ou de se retornar à oficina para reinstrumentalizar o veículo ou trocar os cabos pode ser evitada.

Portanto, propõe-se aqui uma comunicação sem fio a fim de simplificar a instrumentação para o monitoramento da vibração torsional de *powertrain* automotivo.

Como base teórica para a aplicação da tecnologia de comunicação sem fio em problemas de vibração de sistemas rotativos servem de exemplo os trabalhos de Elnady, Sinha e Oyadiji (2012) e Mileyville et al. (2012). A extensão da aplicação dessa tecnologia a outros campos é motivada pelos trabalhos de Whelan et al. (2009) e Qin e Hu (2012).

Metodologicamente, foram empregados sensores magnéticos do tipo *pick-up* para a medição da velocidade de rotação. A comunicação sem fio foi realizada através de módulos

de comunicação EPOSMote II e o sistema como um todo foi submetido a testes em marcha lenta e de tração.

Para a validação deste trabalho foram feitos testes em duas empresas diferentes. Em uma fabricante de embreagens, o sistema desenvolvido foi comparado ao equipamento normalmente usado por essa empresa para a medição da vibração torsional. Em uma montadora, a medição da velocidade de rotação do motor foi validada com os dados provenientes da linha *Controller Area Network* (CAN) do veículo.

OBJETIVOS

Objetivo Geral

Implementar uma medição sem fio da vibração torsional em *powertrain* automotivo a fim de simplificar o procedimento e a manutenção nas atividades de instrumentação e aquisição de sinais.

Objetivos Específicos

- Integrar sensores magnéticos de rotação comerciais com um *hardware* de comunicação sem fio;
- Implementar um *software* para comunicação entre os *hardwares* (transmissor e receptor);
- Testar o funcionamento do sistema em veículo de passeio;
- Comparar os resultados obtidos com sistemas comerciais.

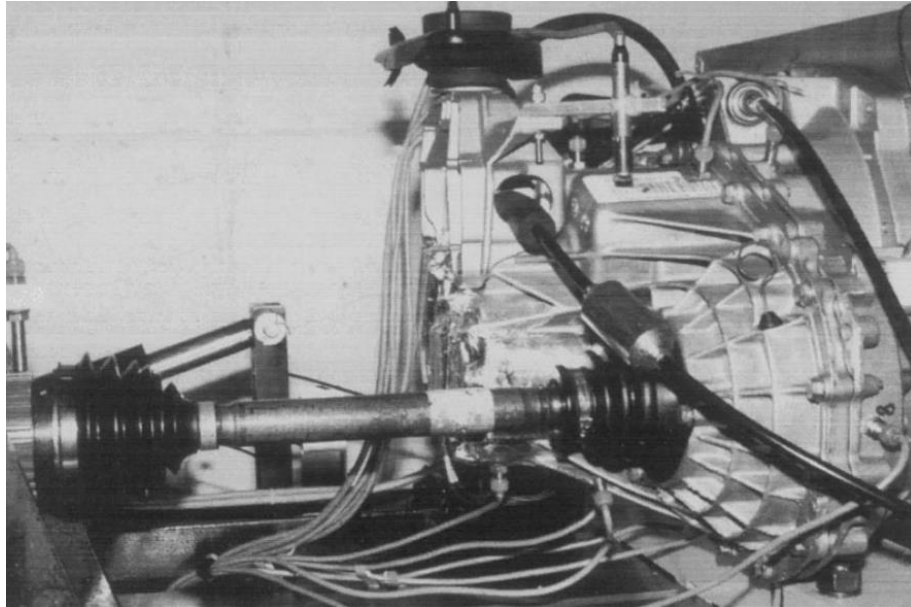
2. REFERENCIAL TEÓRICO

Trabalhos datados da década de 1990 sobre a medição da vibração em eixos rotativos não apresentavam o uso da comunicação sem fio. A exemplo disso, Wang et al. (1992) desenvolveram um sistema capaz de medir simultaneamente a vibração torsional, vertical e horizontal em qualquer posição ao longo de um eixo em rotação. O sistema inclui um módulo óptico, um condicionador de sinal, um módulo de aquisição/transmissão de dados e um pacote de *software* para análise de dados. Um feixe de luz incide sobre uma fita composta por faixas brancas e pretas e colada no eixo em movimento. A luz refletida pela fita é guiada por lentes e por um cabo de fibra óptica até o sensor óptico, que gera uma tensão proporcional à intensidade da luz refletida. Esse sistema não fez uso de comunicação sem fio.

Sarkar, Ellis e Moore (1997) realizaram um experimento para comprovar sua modelagem a cerca de folgas em mecanismos com engrenagens. Eles empregaram um cilindro vazado de alumínio instrumentado com quatro *strain gauges* montados de maneira diametralmente oposta (com dois *strain gauges* de cada lado) para medir o torque no eixo motriz de um acoplamento com duas engrenagens. Essa experimentação também não envolveu transmissão de dados sem fio.

Courdec et al. (1998) projetou uma bancada de testes para simular o fenômeno torsional que ocorre no *driveline* de um veículo sujeito a um torque de 0 a 200Nm. Isso permitiu o estudo da vibração torsional do *driveline* sujeito a torque estático e excitado por flutuações de torque vindas do motor. Em termos de instrumentação, foram inseridos transdutores de torque na caixa de transmissão para medir os torques estático e dinâmico e a velocidade instantânea. A velocidade angular de todas as engrenagens na transmissão foi medida por pontas indutivas apontadas para os dentes das engrenagens. Um acelerômetro foi fixado na carcaça da transmissão para medir as vibrações provenientes de impactos nas engrenagens. Adicionalmente, a temperatura do lubrificante foi controlada e regulada durante os testes. A transmissão instrumentada é mostrada na Figura 1 e nota-se a passagem de muitos cabos.

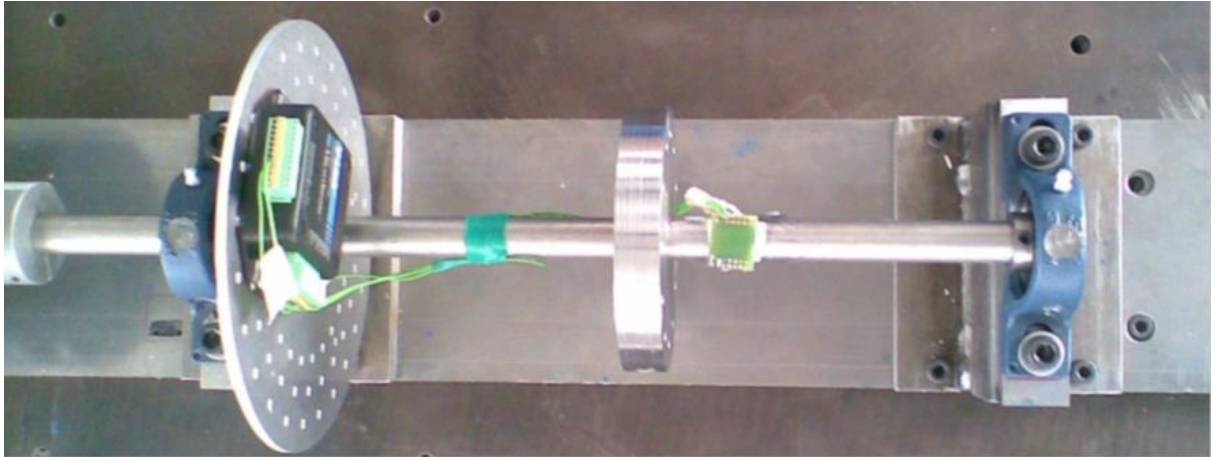
Figura 1 – Transmissão instrumentada em uma bancada de testes.



Fonte: Courdec et al. (1998).

Em contrapartida, trabalhos mais recentes apontam o emprego da transmissão de dados sem fio na medição da vibração em eixos. É o caso do trabalho de Elnady, Sinha e Oyadiji (2012), que propuseram uma metodologia diferente para a medição de velocidades críticas em máquinas rotativas. Ao invés de adotar a metodologia convencional, que mede a vibração em três direções (vertical, horizontal e axial) em cada mancal de rolamento que suporta o rotor de uma máquina rotativa, eles inseriram um único sensor montado no rotor da máquina. Um acelerômetro muito pequeno e leve que adota a tecnologia *Micro Electro Mechanical System* (MEMS) com um módulo *wireless* foi montado diretamente no rotor da máquina. Seu trabalho mostrou que o número de sensores empregados e o custo associado com manutenção podem ser reduzidos significativamente com esse novo conceito. Além disto, os dados da medição adquiridos com esta metodologia podem conter mais informações sobre a vibração do rotor, o que melhora o processo de diagnose. O experimento realizado é ilustrado pela Figura 2, observa-se a presença do módulo *wireless* montado no disco que se situa próximo ao mancal da esquerda.

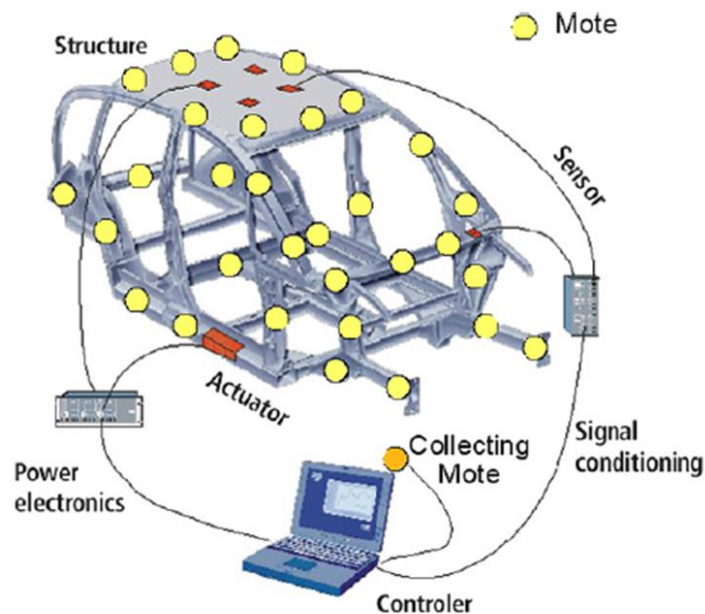
Figura 2 – Experimento para a medição da vibração em um eixo.



Fonte: Elnady, Sinha e Oyadiji (2012).

Mieyeville et al. (2012) sugeriram uma nova solução técnica para o controle ativo da vibração da estrutura de um veículo através da substituição de redes convencionais de sensores com fio por sensores inteligentes conectados sem fio. Sensores piezoelétricos distribuídos na estrutura de um veículo foram conectados a um controlador e regras de controle foram utilizadas para comandar atuadores piezoelétricos posicionados em diferentes locais da carroceria do veículo, como mostrado na Figura 3. Segundo Mieyeville et al. (2012), o monitoramento sem fio surgiu no ano de 2011 como uma tecnologia promissora devido à eliminação de várias restrições de *hardware* e à possibilidade de novas implementações em componentes mecânicos. Além disto, a instalação de poder computacional em sensores abre novas perspectivas no domínio da mecânica, onde usualmente é preferida uma abordagem centralizada.

Figura 3 – Rede de sensores sem fio aplicada para o controle ativo da vibração da estrutura de um veículo.



Fonte: Mieveville et al. (2012).

Além do uso no setor automotivo, a tecnologia de comunicação sem fio vem sendo empregada em outros campos. Como exemplo, Qin e Hu (2012) desenvolveram um tipo de sensor *wireless* embarcado capaz de adquirir sinais de vibração com a finalidade de detectar falhas em caixas de engrenagens planetárias de helicópteros.

Outro exemplo de aplicação em outros campos é o trabalho de Whelan et al. (2009), que demonstrou que a tecnologia de sensoriamento sem fio evoluiu a ponto de se tornar viável e comparável a procedimentos de medição com cabo para a análise modal de grandes estruturas civis. Uma rede de sensores sem fio de larga escala foi empregada para o monitoramento da vibração de uma ponte. O comportamento dinâmico da estrutura foi medido com acelerômetros para a determinação experimental das frequências naturais, dos coeficientes de amortecimento e dos fatores de forma da ponte.

Portanto, nota-se uma tendência recente no uso da comunicação sem fio no monitoramento da vibração de máquinas e estruturas rotativas e essa tecnologia pode ser utilizada em problemas de *powertrain* automotivo.

3. METODOLOGIA

3.1. Condições de teste

A vibração torsional nos elementos do *powertrain* de um veículo pode ser quantificada através da medição da velocidade de rotação de cada componente (volante, engrenagem, eixo, etc). Como cada parte vibra com diferentes amplitudes, é necessário medir os componentes de interesse independentemente, ou seja, com diversos sensores simultaneamente. Para o controle da vibração torsional do *powertrain*, em geral, tomam-se dois pontos de medição: o motor e a entrada da transmissão. Estes representam, respectivamente, a excitação encontrada antes da embreagem e a resposta após a embreagem, pelas quais é possível avaliar a efetividade da redução da excitação do motor após passar pelo amortecimento torsional da embreagem (o ruído de *rattle* pode ser controlado pela rigidez torsional da embreagem). Usualmente, as medições são realizadas a partir de sensores posicionados apontando para a cremalheira do volante do motor e para a engrenagem solidária da transmissão, no eixo piloto ou secundário (LECHNER; NAUNHEIMER; RYBORZ, 1999).

Dentre os diversos ruídos existentes na transmissão, o ruído de embreagem desacoplada (*idle gear noise*) é notado como um ruído desagradável e os consumidores frequentemente o associam de forma incorreta com problemas no motor. Esse ruído é gerado por impactos entre engrenagens que podem aparecer tanto em condições de tração (*rattle noise*), quanto em posição de neutro (*neutral rattle noise*) (DION et al., 2009). Esse ruído é mais acentuado quando a transmissão se encontra na posição de neutro, pois as engrenagens não estão sujeitas a cargas e, conseqüentemente, as folgas entre elas tornam-se maiores (SINGH; XIE; COMPARIN, 1989). Portanto, os testes serão realizados em duas condições: em tração, situação em que se acelera o veículo e que se requisita torque do motor, e em posição de neutro, também conhecida como condição de marcha lenta.

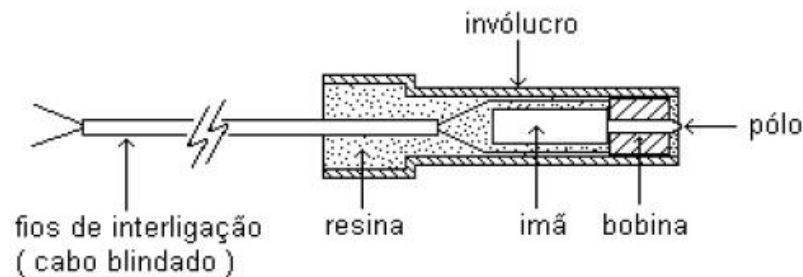
3.2. Sensores magnéticos do tipo *pick-up*

Para a medição da vibração torsional foram escolhidos sensores magnéticos do tipo *pick-up*, pois esse tipo de sensor é largamente utilizado usado na indústria para realizar tal medição. Adicionalmente, existem seis tipos de sensores de velocidade angular que são comumente usados em sistemas automotivos. Dentre eles, os sensores magnéticos são de

baixo custo, possuem dimensões pequenas a moderadas, não precisam de fontes externas para gerar seus sinais e não são muito sensíveis a temperatura. Por outro lado, apresentam perda de sinal a velocidade zero. A amplitude e a fase do sinal dependem da velocidade de rotação da respectiva engrenagem acoplada e a instalação é geralmente limitada por uma distância de cerca de 2mm entre sensor e engrenagem (FLEMING, 2001).

Esse tipo de sensor tem a finalidade de gerar uma tensão pulsante quando um material ferromagnético passa em frente ao seu polo. Basicamente, ele consiste em um ímã permanente e uma bobina alojados dentro de um invólucro de material não ferromagnético, como ilustrado na Figura 4.

Figura 4 – Diagrama esquemático de um sensor magnético do tipo *pick-up*.

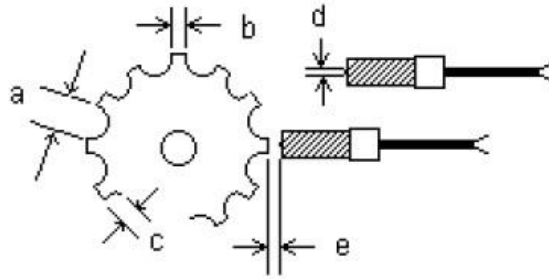


Fonte: Turbtron Indústria e Comércio de Equipamentos Elétricos Ltda (2001).

De acordo com a Lei de Faraday, uma variação entre um campo magnético (gerado pelo ímã) e um condutor de energia elétrica (bobina) induz uma tensão neste. Essa tensão induzida é alternada, pulsante e com uma frequência proporcional à velocidade do material ferromagnético que passa em frente ao polo do sensor (TURBTRON INDÚSTRIA E COMÉRCIO DE EQUIPAMENTOS ELÉTRICOS LTDA, 2001).

Geralmente, utiliza-se uma engrenagem como material ferromagnético e algumas restrições devem ser obedecidas para a instalação do sensor, como mostrado na Figura 5. A amplitude do sinal de saída é função principalmente da distância entre o polo do sensor e o dente da engrenagem e da velocidade angular da engrenagem (TURBTRON INDÚSTRIA E COMÉRCIO DE EQUIPAMENTOS ELÉTRICOS LTDA, 2001).

Figura 5 – Restrições para a instalação de um sensor magnético do tipo *pick-up*.



Fonte: Turbtron Indústria e Comércio de Equipamentos Elétricos Ltda (2001).

Em que,

$$d = \text{diâmetro do polo do sensor} \quad (1)$$

$$a > 3d \quad (2)$$

$$b > 2d \quad (3)$$

$$c > 2d \quad (4)$$

$$0,1\text{mm} \leq e \leq 0,5\text{mm} \quad (5)$$

$$\text{Espessura da engrenagem} > 2d \quad (6)$$

Um impulso elétrico na saída é gerado toda vez que um dente da engrenagem passa em frente ao polo do sensor. Já que a engrenagem consiste em uma sequência de dentes, é gerada na saída uma sequência de impulsos elétricos com frequência diretamente proporcional ao número de dentes da engrenagem e à velocidade angular da mesma. Essa relação é dada pela Equação 7.

$$f = \frac{\omega_{\text{RPM}} \cdot n}{60} \quad (7)$$

Em que, f é a frequência dos impulsos elétricos expressa em Hz, ω_{RPM} é a velocidade angular da engrenagem em rotações por minuto e n é o número de dentes da engrenagem.

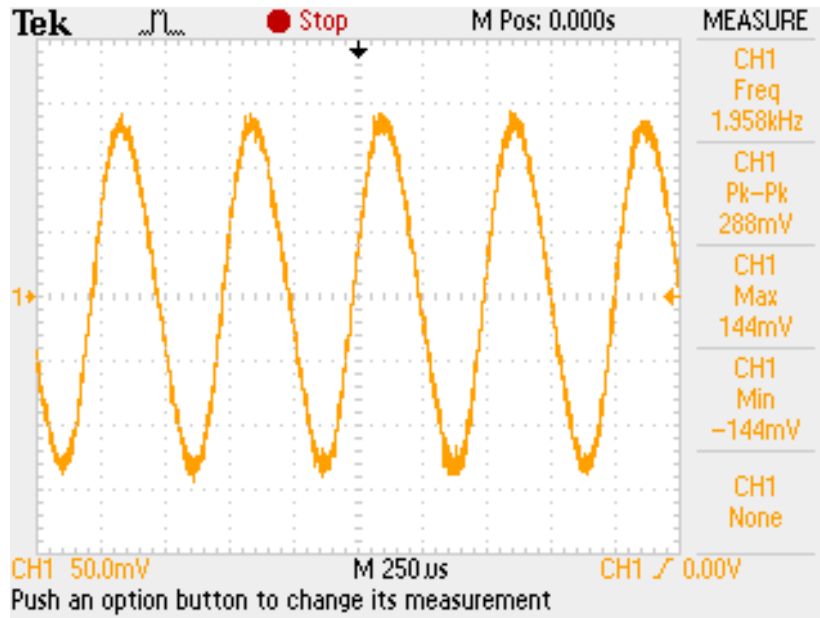
Em relação às aplicações desse tipo de sensor, Turbtron Indústria e Comércio de Equipamentos Elétricos Ltda (2001) aponta que:

Os sensores magnéticos são utilizados para enviar sinais para medição e controle de velocidade de equipamentos tais como: tacômetros, velocímetros, controle de velocidade de motores automotivos, motores estacionários, reguladores de velocidade, turbinas hidráulicas, turbinas à vapor, ventiladores, exaustores, centrífugas, sistemas de freios ABS e em várias outras aplicações da área industrial, mesmo em condições severas de temperatura.

A fim de se avaliar o sinal gerado por esse tipo de sensor, foi realizado um teste com o mesmo apontado para uma engrenagem acoplada no eixo de um torno. A velocidade de

rotação do torno foi fixada em 2000rpm e o número de dentes da engrenagem era 58. Um osciloscópio modelo TDS 2002C da fabricante Tektronix foi usado para gravar o sinal gerado pelo sensor, que é apresentado a seguir na Figura 6.

Figura 6 – Sinal gerado pelo sensor magnético do tipo *pick-up* em teste realizado em torno.



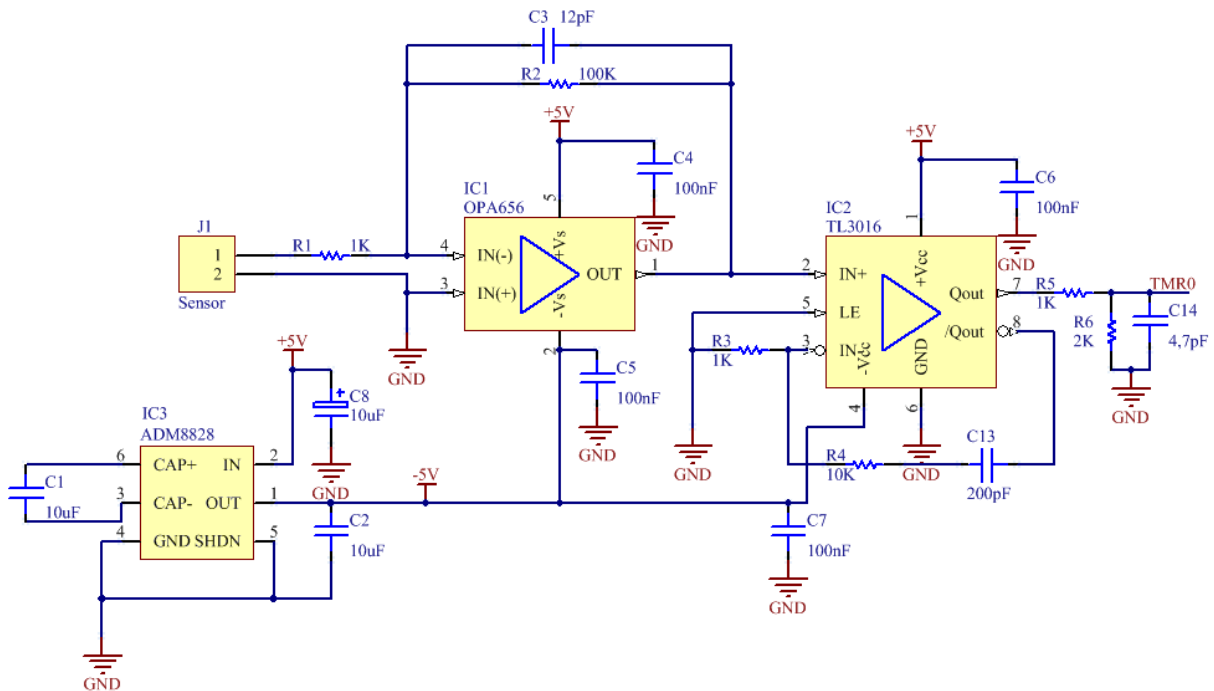
Fonte: autor.

Nota-se que o sinal gerado é uma onda senoidal centrada em zero com amplitude de cerca de 290mV. Uma vez que esse sinal não pode ser aplicado diretamente na entrada de um microcontrolador, é necessário transformá-lo em um trem de pulsos e para essa função foram confeccionadas placas de condicionamento de sinal, que são tratadas no próximo subcapítulo.

3.3. Placas de condicionamento de sinal

Conforme mostrado na seção anterior pela Figura 6, o sensor magnético do tipo *pick-up* gera um sinal senoidal com amplitude da ordem de 290mV. Esse sinal deve ser amplificado e em seguida transformado em um trem de pulsos, ou seja, em um sinal *Transistor-Transistor Logic* (TTL) para que então esse possa alimentar a entrada do microcontrolador. Com isso em vista, foram desenvolvidas placas de condicionamento de sinal e cada placa consiste principalmente em um amplificador operacional e em um comparador. Uma parte do diagrama esquemático dessa placa é apresentada na Figura 7.

Figura 7 – Parte do diagrama esquemático da placa de condicionamento de sinal com ênfase no amplificador operacional e no comparador.



Fonte: autor.

O diagrama esquemático foi elaborado com o auxílio do *software* Altium Designer. O amplificador operacional selecionado, componente IC1 na Figura 7, foi o OPA656 da fabricante Texas Instruments. Ele foi montado na configuração inversora com ganho de -100. Por sua vez, para o comparador, componente IC2 na Figura 7, foi escolhido o TL3016 também da fabricante Texas Instruments. A tensão de saída de nível lógico baixo desse componente vale tipicamente 500mV, ao passo que o valor típico da tensão de saída de nível lógico alto é 3,9V. Como o módulo de comunicação EPOSMoteII admite tensões de entrada de no máximo 3,3V, foi adicionado um divisor de tensão na saída do comparador composto pelos resistores R5 e R6 que diminui a tensão de saída para 2,6V. Além da tensão de alimentação positiva, o amplificador operacional requer uma tensão de alimentação de -5V. Para solucionar isso, foi incluído o ADM8828 da fabricante Analog Devices, componente IC3 na Figura 7. Trata-se de um inversor de tensão que usa a técnica do capacitor chaveado. Foram incluídos também capacitores de desacoplamento, componentes C4, C5, C6 e C7 na Figura 7, para filtrar a tensão de alimentação do amplificador operacional e do comparador.

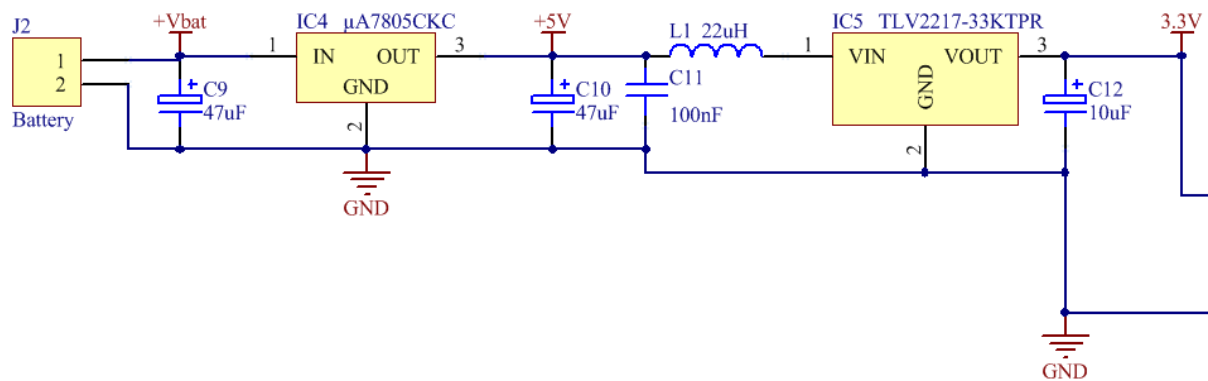
Os componentes R3, R4 e C13 servem para a realimentação do comparador e possuem a função de diminuir o ruído nas transições, ou seja, eles atuam como uma espécie de *debouncer*. O valor de R3 foi fixado em 1kΩ e os valores de R4 e C13 foram determinados

por cerca de uma ordem de grandeza maior que a frequência máxima de leitura ($f_{\text{máx}}$), como descrito pela Equação 8.

$$10. f_{\text{máx}} = \frac{1}{2\pi \cdot R4 \cdot C13} \quad (8)$$

Outra função da placa de condicionamento de sinal é fornecer energia para o módulo de comunicação EPOSMoteII durante a aplicação. Nesse caso, a tensão de alimentação deve ser de 2,0 a 3,6VDC. A parte do diagrama esquemático que contém os reguladores de tensão é apresentada na Figura 8.

Figura 8 – Parte do diagrama esquemático da placa de condicionamento de sinal com ênfase nos reguladores de tensão.

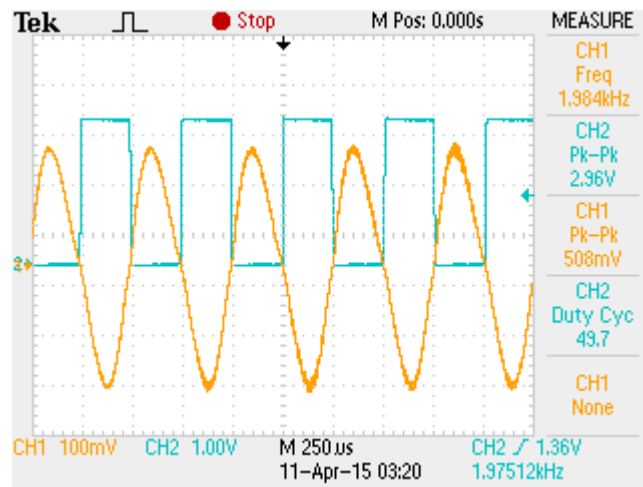


Fonte: autor.

A partir de uma bateria de 9V, deve-se obter uma tensão de 5V para alimentar o amplificador operacional e o comparador, bem como uma tensão entre 2,0 e 3,6V para alimentar o EPOSMoteII. O componente IC4 na Figura 8 é um regulador de tensão da fabricante STMicroelectronics, ele fornece uma tensão de saída fixa em 5V a partir de uma tensão de entrada de até 35V. Já o componente IC5 na Figura 8 é um regulador de tensão da fabricante Texas Instruments que fornece uma tensão de saída de 3,3V a partir de uma tensão de entrada de 3 a 12V.

Com o intuito de avaliar o funcionamento da placa de condicionamento de sinal, um novo teste foi realizado com o sensor magnético apontado para uma engrenagem de 58 dentes fixa no eixo de um torno girando a 2000rpm. O resultado desse teste é mostrado em seguida pela Figura 9.

Figura 9 – Teste realizado para avaliar o funcionamento da placa de condicionamento de sinal.



Fonte: autor.

O sinal senoidal gerado pelo sensor magnético no canal 1 foi transformado em um sinal TTL, conforme observado no canal 2. Como foi empregado um amplificador operacional na configuração inversora, os semiciclos negativos do sinal do sensor foram transformados em níveis lógicos altos e os semiciclos positivos equivalem a níveis lógicos baixos. Além disso, a amplitude de pico a pico foi aumentada de 508mV para 2,96V e o sinal de saída da placa de condicionamento de sinal pode ser aplicado no *timer* do EPOSMoteII.

3.4. Módulos de comunicação EPOSMoteII

O módulo de comunicação a ser utilizado é o EPOSMoteII, visto que esse módulo foi desenvolvido na Universidade Federal de Santa Catarina pelo Laboratório de Integração *Software/Hardware* (LISHA), que participa deste projeto.

Em termos de *software*, o *Embedded Parallel Operating System* (EPOS), também desenvolvido pelo LISHA, permite a automação do projeto de sistemas embarcados de modo que os projetistas possam se concentrar na aplicação em si. Razão para isso é que o EPOS se baseia no método *Application-Driven Embedded System Design* (ADESD) (LISHA, 2015).

Do ponto de vista de *hardware*, o projeto EPOSMote foi inicialmente concebido para o desenvolvimento de um módulo para comunicação sem fio entre sensores baseado na plataforma ZigBee e focado em aplicações para a agricultura de precisão. A primeira versão, a EPOSMoteI, é baseada no módulo MeshNetics ZigBit MNZB-24-Bo, com um controlador

AVR de 8 bits, comunicação de acordo com o padrão IEEE 802.15.4 e com um pequeno conjunto de sensores (temperatura, humidade, etc) (LISHA, 2015, tradução nossa).

A segunda versão do *hardware*, a EPOSMoteII, visa a modularização do projeto EPOSMote. O intuito é oferecer módulos diferentes e intercambiáveis para cada função, como processamento/comunicação, sensoriamento e fornecimento de energia. Uma comparação entre as versões EPOSMoteI e EPOSMoteII é retratada na Figura 10.

Figura 10 – Comparação entre as versões EPOSMoteI (a) e EPOSMoteII (b).



Fonte: LISHA (2015).

Para o módulo de processamento/comunicação foi selecionado o microcontrolador MC13224V da fabricante Freescale. Ele é baseado em um processador ARM7 de 32 bits e possui um rádio integrado, 128Kbyte de memória flash, 80Kbyte de memória ROM e 96Kbyte de memória RAM. Um chip de antena conectado ao rádio do microcontrolador MC13224V e um oscilador externo de cristal de 24MHz integram esse módulo (LISHA, 2015).

Por sua vez, o módulo de sensoriamento tem a função de equipar o EPOSMoteII com sensores e atuadores. Ele se comunica com o módulo de processamento/comunicação através de interfaces UART ou SPI e envia sinais analógicos pelos pinos com conversores ADC. Além disso, possui também uma interface USB para UART, o que viabiliza a comunicação com um PC. Um módulo de sensoriamento chamado Startup Board foi desenvolvido no âmbito do projeto EPOSMote e contém um acelerômetro de 3 eixos, um sensor de temperatura, 2 botões e 3 LEDs. Adicionalmente, um módulo de fornecimento de energia que controla mais de uma fonte pode ser conectado ao módulo de processamento/comunicação (LISHA, 2015).

O padrão IEEE 802.15.4 define o protocolo e compatibilidade de interconexão para dispositivos de comunicação que usam baixa taxa de transmissão de dados, pequeno consumo

de energia e transmissões de rádio frequência (RF) de pequeno alcance em uma área de rede sem fio pessoal (WPAN) (IEEE COMPUTER SOCIETY, 2011). Esse padrão é a base para as especificações ZigBee e segundo Nolte, Hansson e Bello, conforme indicado por Mieveville (2012), dentre as soluções para comunicação sem fio que podem ser usadas no setor automotivo (Bluetooth, Ultra Wide Band e Wi-Fi), a ZigBee é considerada uma boa solução, embora não seja muito apropriada para aplicações de tempo real.

3.5. Projeto e desenvolvimento do *software*

O módulo de comunicação EPOSMoteII é programado em C++ em ambiente Linux. Sua programação foi dividida em duas partes principais: programa para leitura do sensor magnético e programa para envio de mensagens.

3.5.1. Leitura do sensor magnético

O objetivo aqui é obter a frequência dos pulsos gerados pelo sensor magnético com o menor erro relativo de leitura possível. Três programas diferentes foram escritos para tanto e a escolha do programa mais adequado foi feita baseada em uma comparação do erro relativo de cada programa em uma faixa de 1000 a 7000 Hz. Em seguida, o programa selecionado foi otimizado para operar nas faixas de frequência esperadas durante os testes em veículo.

3.5.1.1. Programa 1

O primeiro programa faz uso de um *timer* e de um alarme. O *timer* deve gerar uma interrupção cada vez que certa quantidade de pulsos for lida. Essa quantidade de pulsos é especificada pelo parâmetro *comp1* do *timer*, ou seja, um parâmetro usado para fazer comparação. Durante a interrupção do *timer* um contador global é incrementado. A lógica que envolve esse programa é ler a cada período do alarme o valor desse contador global. O trecho do programa 1 que contém as funções do *timer* e do alarme é mostrado na Figura 11.

Figura 11 – Trecho do programa 1 com as funções do *timer* e do alarme.

```
39 void alarm_func(void)
40 {
41     static unsigned int value = 0;
42     cout << (count - value) << endl;
43     value = count;
44 }
45
46 void timer_func(void)
47 {
48     count++;
49 }
```

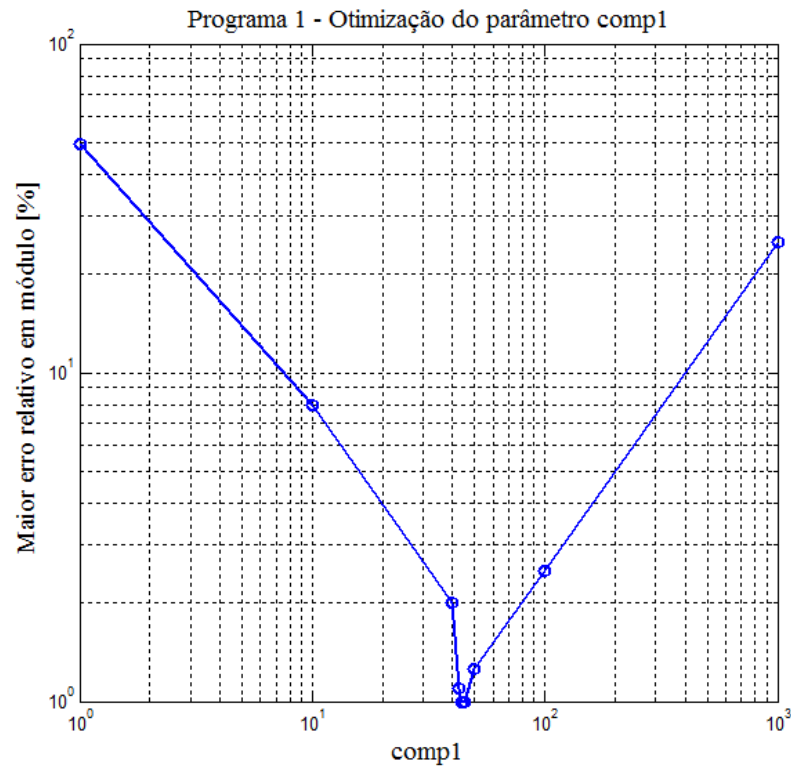
Fonte: autor.

A frequência dos pulsos lidos (f) é dada pela seguinte expressão,

$$f = \frac{\text{comp1} \cdot (\text{count} - \text{value})}{\text{ALARM_PERIOD}} \quad (9)$$

Pode-se perceber que existem dois parâmetros a serem otimizados no programa 1: `comp1` e `ALARM_PERIOD`. Num primeiro momento, otimizou-se o valor de `comp1`. A frequência dos pulsos foi fixada em 4000Hz, pois esse é o valor médio da faixa de frequências de 1000 a 7000Hz considerada para a comparação dos três programas, e o período do alarme foi mantido em 1s. Assim, variou-se o valor de `comp1` de forma a se obter o menor erro relativo de leitura possível. O resultado dessa otimização é apresentado na Figura 12.

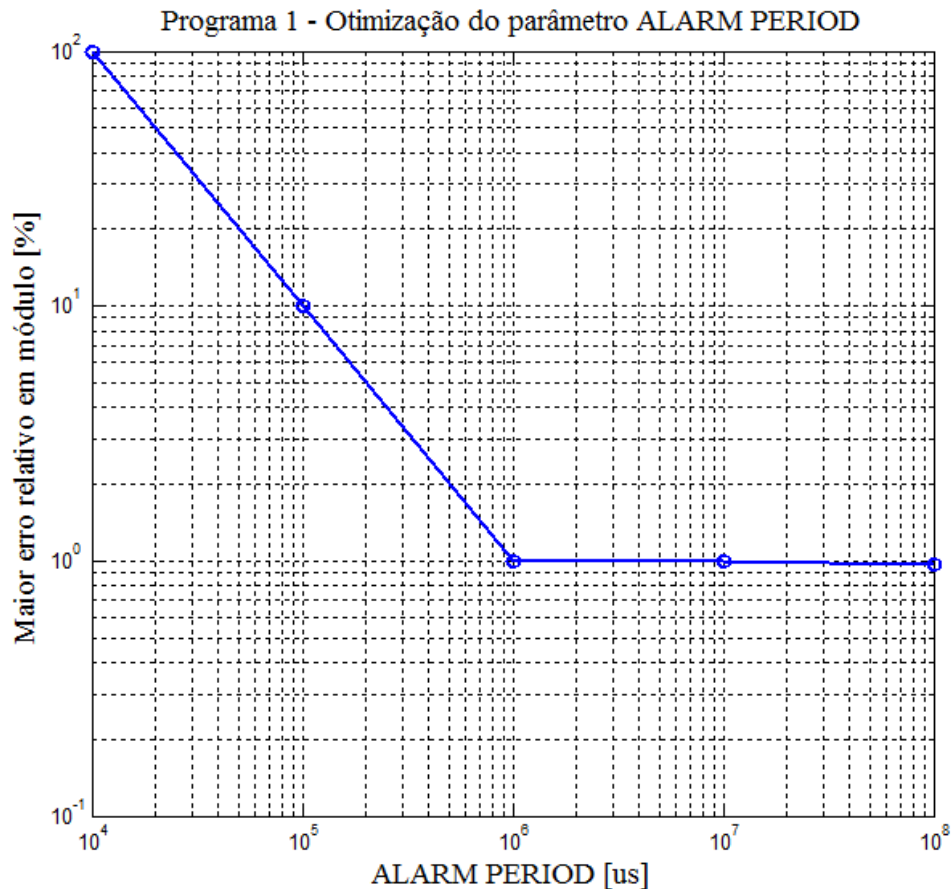
Figura 12 – Programa 1 – Otimização do parâmetro comp1.



Fonte: autor.

Nota-se que o erro relativo de leitura não é inversamente proporcional ao valor de comp1, mas que há um valor ótimo de comp1 para o qual o erro relativo é mínimo. De fato, o menor erro relativo alcançado foi 1,000% e o correspondente valor de comp1 é 44. Em seguida, otimizou-se o parâmetro ALARM_PERIOD com a frequência dos pulsos mantida em 4000Hz e agora com comp1 igual a 44. O resultado da segunda otimização é tratado pela Figura 13.

Figura 13 – Programa 1 – Otimização do parâmetro ALARM_PERIOD.



Fonte: autor.

Pode-se perceber que agora sim o erro relativo é inversamente proporcional ao período do alarme. Para um período de 100s o erro relativo obtido foi de 0,967%, o que leva a conclusão de que deve existir uma assíntota para o erro relativo muito próxima desse valor. Entretanto, períodos de alarme muito grandes são impraticáveis.

3.5.1.2. Programa 2

Ao contrário do programa 1, o programa 2 não trabalha com interrupções do *timer*. Por isso, o parâmetro *comp1* deve ser configurado em seu valor máximo, ou seja, 0xFFFF. A ideia aqui é ler periodicamente o valor do próprio contador do *timer*, o *cntr*. Desse modo, a função do alarme lê diretamente o valor de *cntr* e em seguida zera o mesmo. Já a função do *timer* apenas informa que o valor de *comp1* foi alcançado, o que ocorre quando 65535 pulsos são lidos. O trecho do programa 2 que exhibe as funções do *timer* e do alarme é apresentado em sequência na Figura 14.

Figura 14 – Trecho do programa 2 com as funções do *timer* e do alarme.

```

39 void alarm_func(void)
40 {
41     unsigned int value = timer -> read();
42     timer -> reset();
43     cout << value << endl;
44 }
45
46 void timer_func(void)
47 {
48     cout << "Limite" << endl;
49 }

```

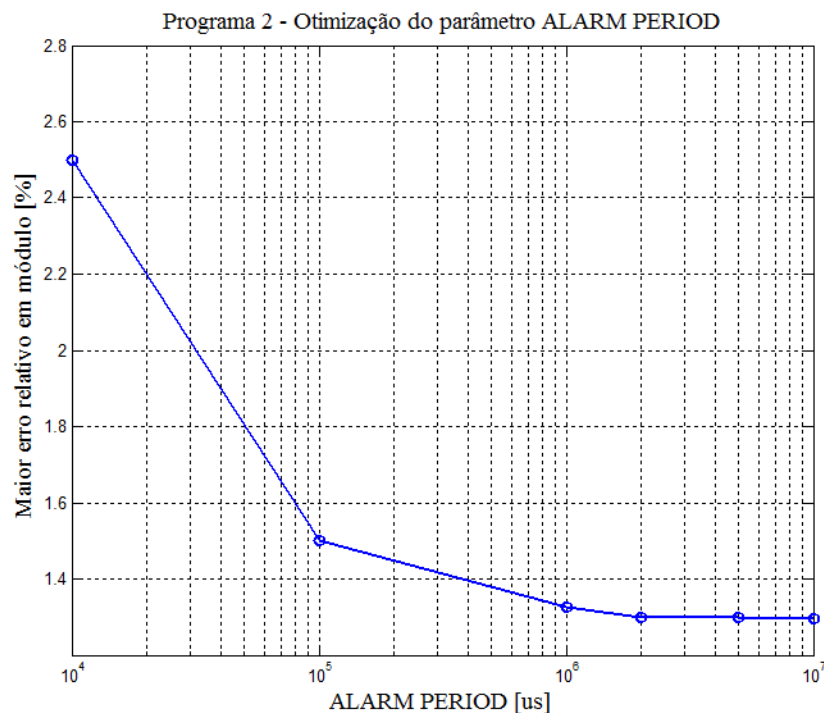
Fonte: autor.

A frequência dos pulsos lidos (f) é dada pela seguinte expressão,

$$f = \frac{\text{value}}{\text{ALARM_PERIOD}} \quad (10)$$

Claramente, existe apenas um parâmetro a ser otimizado no programa 2 e esse é o ALARM_PERIOD. Com a frequência dos pulsos mantida em 4000Hz, variou-se seu valor a fim de se encontrar o menor erro relativo de leitura. O resultado dessa otimização é apresentado na Figura 15.

Figura 15 – Programa 2 – Otimização do parâmetro ALARM_PERIOD.



Fonte: autor.

De forma semelhante ao que ocorreu com o programa 1, o erro relativo mostrou-se inversamente proporcional ao período do alarme. Para um período de 1s, obteve-se um erro de 1,325% e supõe-se novamente a existência de uma assíntota, dessa vez em torno de 1,3%. Vale a pena mencionar que com o período do alarme configurado em 1ms não foi possível efetuar a leitura, pois ocorreu a mensagem de erro “Prefetch abort”, que basicamente significa que os dados do *timer* não puderam ser acessados (ARM INFORMATION CENTER, 2015). No outro extremo, com o período do alarme configurado em 100s o limite de `comp1` foi alcançado e um valor absurdo foi lido, motivo pelo qual esse ponto não foi considerado na Figura 15.

3.5.1.3. Programa 3

O programa 3 emprega uma lógica inversa aos programas anteriores. Ao invés de contar o número de pulsos a cada período fixo de tempo, ele conta o tempo a cada número fixo de pulsos lidos. Para tanto, ele faz uso de um cronômetro e a função do *timer* lê o valor desse cronômetro e logo após o reinicia. O trecho do programa 3 que contém a função do *timer* é exibido a seguir na Figura 16.

Figura 16 – Trecho do programa 3 com a função do *timer*.

```

37 void timer_func(void)
38 {
39     value = chrono.ticks();
40     chrono.reset();
41     chrono.start();
42     cout << value << endl;
43 }

```

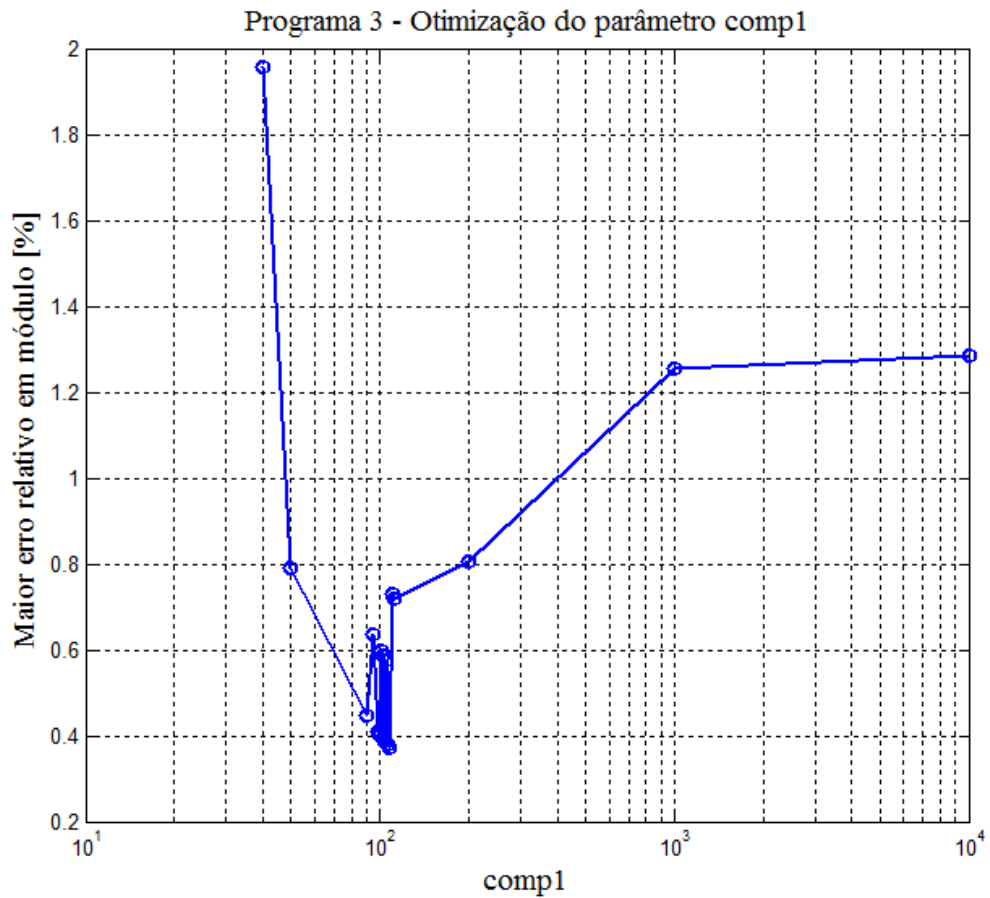
Fonte: autor.

A frequência dos pulsos lidos (f) é dada pela seguinte expressão,

$$f = \text{comp1} \cdot \frac{10000}{\text{value}} \quad (11)$$

A constante 10000 na expressão acima é a frequência de funcionamento do cronômetro, ou seja, um valor de 10000 na leitura do cronômetro significa que 1s se passou. Nota-se que para o programa 3 é necessário otimizar somente o parâmetro `comp1`. Então, variou-se o valor de `comp1` com a frequência dos pulsos mantida em 4000Hz com o intuito de obter-se o menor erro relativo de leitura. O resultado dessa otimização é apresentado na Figura 17.

Figura 17 – Programa 3 – Otimização do parâmetro comp1.

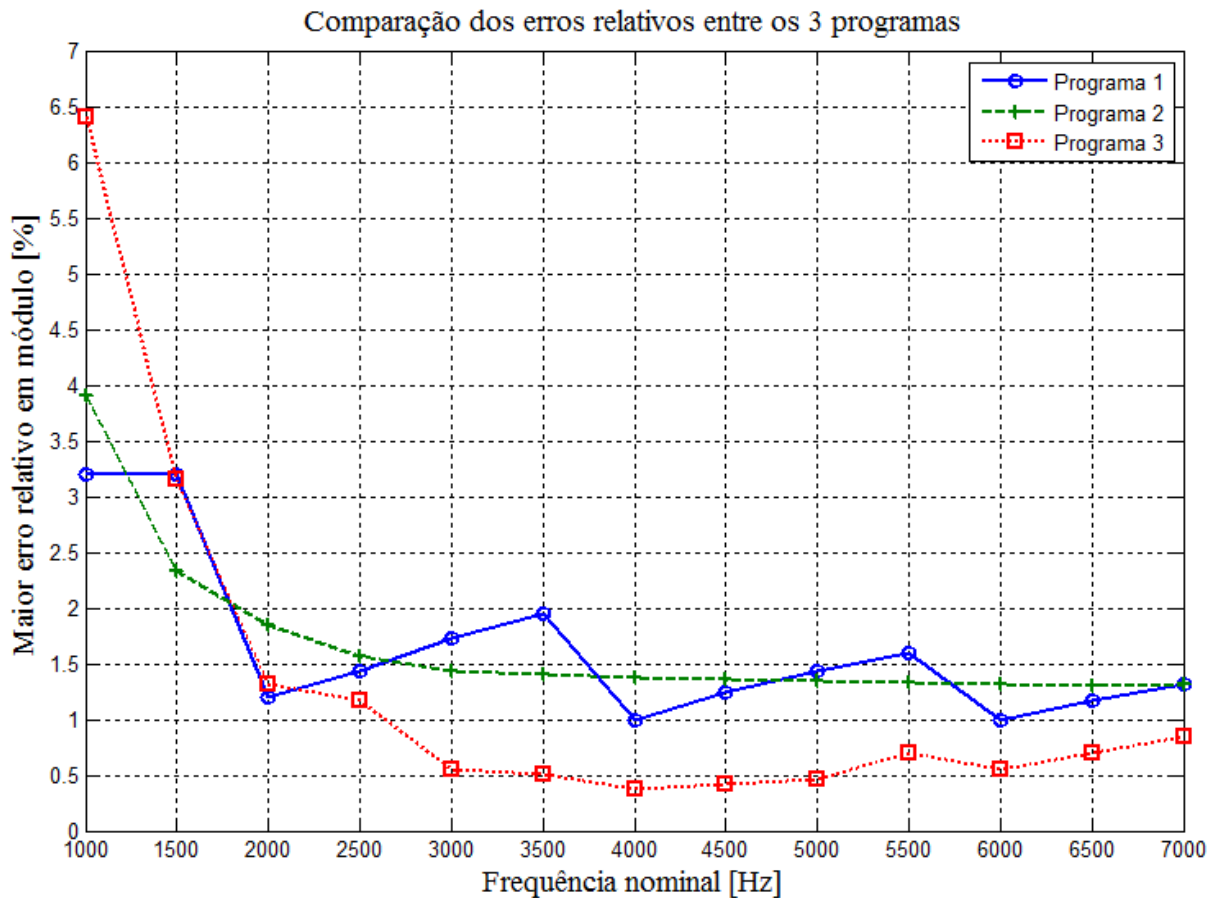


Fonte: autor.

O resultado obtido foi semelhante ao programa 1. Constata-se que o erro relativo não é inversamente proporcional ao número de pulsos lidos, porém há um valor ótimo de comp1 para o qual o erro relativo é mínimo. Para um valor de comp1 igual a 108, obteve-se o menor erro relativo: 0,372%. É importante relatar aqui que houve um erro de leitura para comp1 igual a 10. Logo após o início da execução do programa, alguns valores coerentes foram impressos, mas após isso apenas o valor 0 foi obtido, o que descarta o uso de comp1 com valor menor ou igual a 10.

Com os três programas otimizados para uma frequência de 4000Hz, elaborou-se uma comparação dos erros relativos desses programas para uma faixa de frequências de 1000Hz a 7000Hz com passos de 500Hz. O objetivo agora é selecionar o programa que apresentar o menor erro relativo de leitura. Essa comparação é mostrada pela Figura 18.

Figura 18 – Comparação dos erros relativos entre os 3 programas.



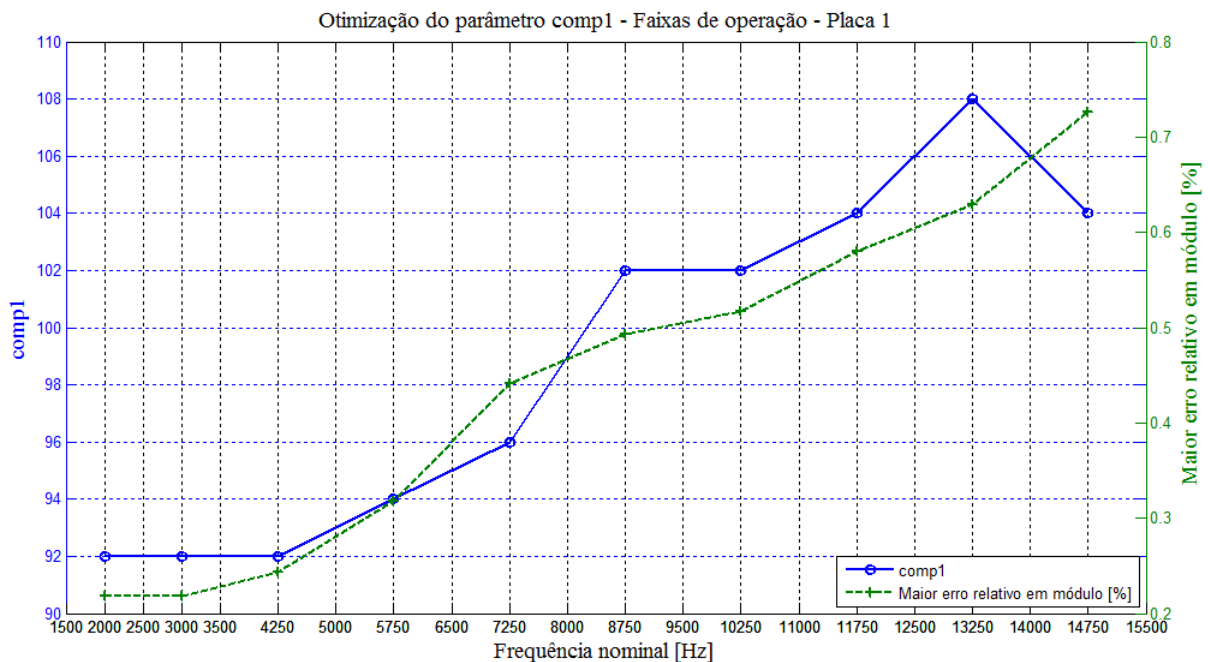
Fonte: autor.

Nota-se que o erro relativo se comporta de maneira diferente para cada programa. No programa 1, o erro relativo oscila com o formato de uma onda. Embora o menor erro obtido tenha sido em 4000Hz, erros muito próximos a esse foram obtidos em frequências de 2000Hz e 6000Hz. Já no programa 2, o erro relativo se mostrou inversamente proporcional à frequência dos pulsos lidos, o que possibilita até mesmo uma previsão do erro. Supõe-se mais uma vez a existência de uma assíntota em torno de 1,3%. Por sua vez, o erro relativo do programa 3 apresentou um mínimo em 4000Hz e cresceu continuamente nos dois sentidos a partir desse ponto. Embora tenha apresentado o maior erro relativo na frequência de 1000Hz, o programa 3 mostrou-se o mais exato ao longo da faixa de frequências avaliada e, por isso, foi o programa escolhido. Uma constatação importante feita aqui é que é mais exato medir o tempo decorrido entre a passagem de um número fixo de pulsos do que medir o número de pulsos dentro de um período fixo de tempo.

3.5.1.4.Otimização do programa 3

Conforme apontado na seção anterior, o único parâmetro a ser otimizado no programa 3 é o comp1. A partir da Figura 18, observa-se que o erro relativo de leitura é mínimo para a frequência na qual o comp1 foi otimizado e cresce nos dois sentidos a partir desse ponto. Logo, deve-se otimizar o comp1 ao longo de toda a faixa de frequências esperada. A primeira faixa de frequências esperada, referente à placa 1, vai de 1500Hz a 15500Hz e considerou-se para isso uma cremalheira do volante do motor com 150 dentes e com velocidade de rotação de 600 a 6200rpm. Essa faixa foi dividida em 10 faixas de otimização para o programa 3, sendo as duas primeiras faixas de otimização com 1000Hz de alcance e as demais com 1500Hz de alcance. Dessa forma, o parâmetro comp1 foi otimizado para frequências no meio de cada uma dessas 10 faixas. O resultado dessa otimização é mostrado abaixo pela Figura 19.

Figura 19 – Programa 3 – Otimização do parâmetro comp1 para uma faixa de frequências de 1500Hz a 15500Hz.



Fonte: autor.

Nota-se que o valor ótimo de comp1 variou de 92 até 108 ao longo da faixa de frequências avaliada e que para algumas faixas de otimização foram obtidas o mesmo valor de comp1. A partir desse resultado é possível definir cinco faixas de operação, conforme a Tabela 1. O único ajuste feito foi para a frequência de 13250Hz, para a qual o valor ótimo de

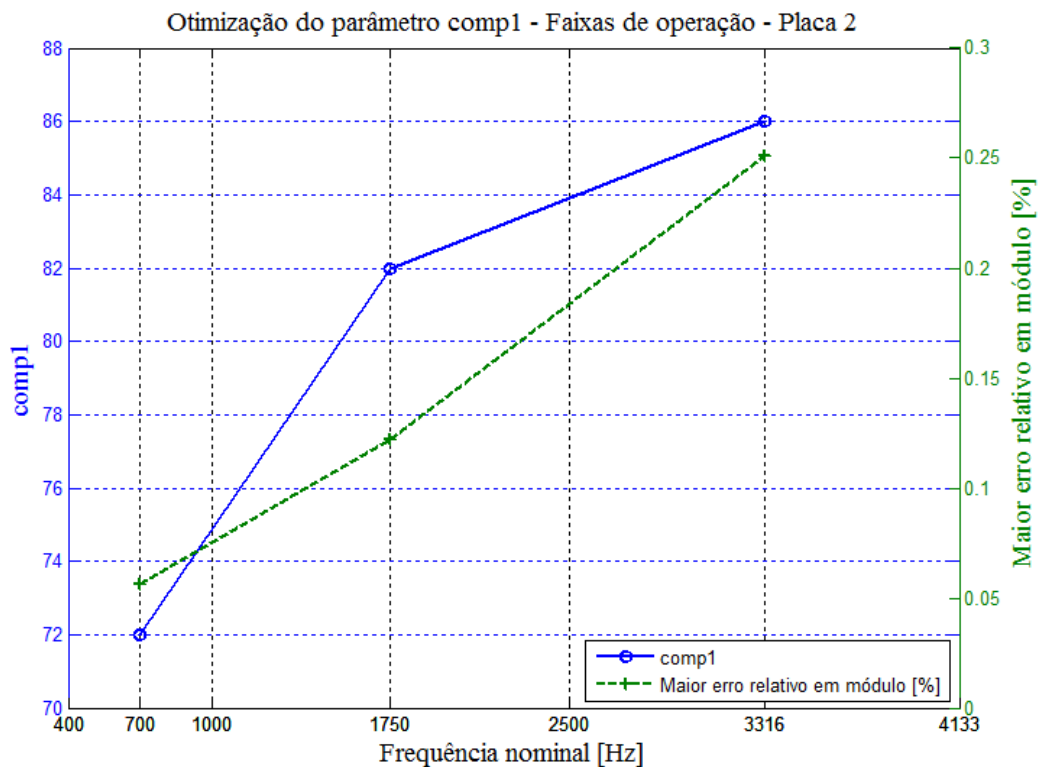
comp1 encontrado foi 108, mas para a definição das faixas de operação considerou-se 104. De fato, o erro relativo aumentou apenas 0,016% com esse ajuste. Como pode ser visto na escala a direita na Figura 19, o erro relativo cresceu continuamente em função da frequência e variou de 0,218% a 0,726%.

Tabela 1 – Faixas de operação da placa 1 e seus respectivos valores de comp1.

Faixa de operação	Frequência [Hz]	comp1
1	$f < 5000$	92
2	$5000 \leq f < 6500$	94
3	$6500 \leq f < 8000$	96
4	$8000 \leq f < 11000$	102
5	$f \geq 11000$	104

A segunda faixa de frequências esperada, referente à placa 2, vai de 400Hz a 4133Hz e considerou-se para isso uma engrenagem no eixo piloto da transmissão com 50 dentes girando de 600 a 6200rpm. Essa faixa foi dividida em 3 faixas de otimização para o programa 3, sendo a primeira com 600Hz de alcance e a segunda com 1500Hz de alcance. Assim, o parâmetro comp1 foi otimizado no meio de cada uma dessas 3 faixas de otimização. O resultado disso é mostrado a seguir na Figura 20.

Figura 20 – Programa 3 – Otimização do parâmetro comp1 para uma faixa de frequências de 400Hz a 4133Hz.



Fonte: autor.

Para a segunda faixa de frequências o valor ótimo de comp1 variou de 72 a 86 e o erro relativo cresceu continuamente de 0,056% a 0,251%. Nesse caso, as faixas de operação tornaram-se iguais às faixas de otimização e a definição delas é registrada na Tabela 2.

Tabela 2 – Faixas de operação da placa 2 e seus respectivos valores de comp1.

Faixa de operação	Frequência [Hz]	comp1
1	$f < 1000$	72
2	$1000 \leq f < 2500$	82
3	$f \geq 2500$	86

Até aqui, considerou-se apenas o resultado de uma amostra. Com o intuito de reduzir o erro relativo de leitura, deve-se considerar a média de algumas amostras. Para tanto, existem basicamente dois métodos para se calcular a média de amostras: por número fixo de amostras ou por tempo. O primeiro método elaborado foi o com médias através de número fixo de amostras. Ele consiste em ler o valor do cronômetro apenas quando o número desejado de

amostras é alcançado, o que foi implementado com uma estrutura de *if* no corpo da função *main*, como ilustrado pela Figura 21.

Figura 21 – Trecho do programa 3 otimizado para ler médias através de número fixo de amostras.

```

30     while(1)
31     {
32         if(count==N_SAMPLES)
33         {
34             value = chrono.ticks();
35             chrono.reset();
36             chrono.start();
37             cout << value << endl;
38             count = 0;
39         }
40     }
41 }
42
43 void timer_func(void)
44 {
45     count ++;
46 }

```

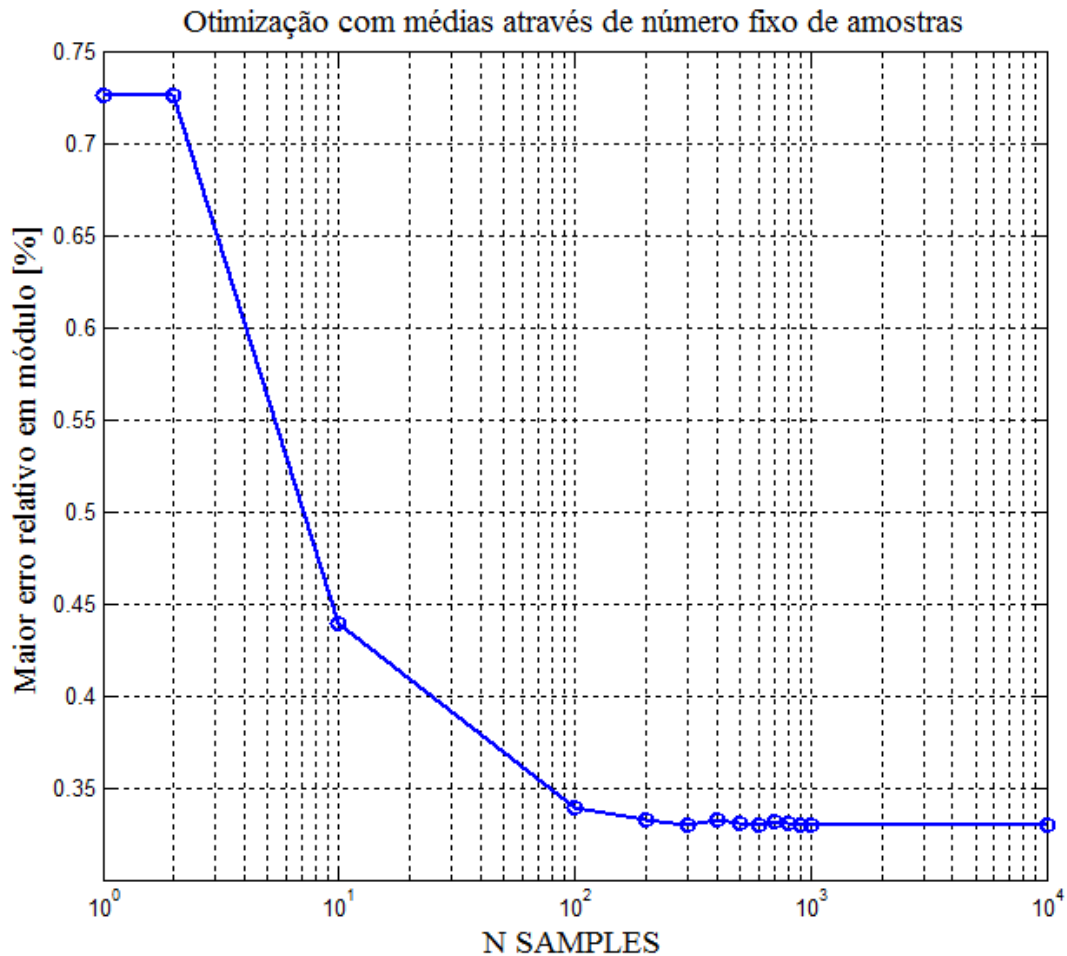
Fonte: autor.

Nesse caso, a frequência dos pulsos lidos (f) é dada pela seguinte expressão,

$$f = N_SAMPLES \cdot \text{comp1} \cdot \frac{10000}{\text{value}} \quad (12)$$

Para avaliar a efetividade dessa alteração no programa 3, considerou-se o pior caso da otimização anterior, ou seja, a frequência de 14750Hz para a qual o erro relativo foi de 0,726%. Com a frequência mantida nesse valor e com *comp1* fixo em 104, variou-se o parâmetro *N_SAMPLES*. O resultado dessa otimização é apresentado na Figura 22.

Figura 22 – Otimização do programa 3 – Médias através de número fixo de amostras.



Fonte: autor.

O erro relativo foi reduzido de 0,726% na condição inicial sem média para 0,329% quando foi calculada a média de 1000 amostras. Portanto, foi possível reduzir o erro para menos da metade. A única desvantagem de se trabalhar com um número grande de amostras é que o período de amostragem se torna impraticável. Novamente, supõe-se a existência de uma assíntota em torno de 0,33%.

O segundo método elaborado para reduzir o erro relativo foi o de média por tempo. Para especificar o período de amostragem foi instituído um alarme. A função do alarme apenas muda o estado de uma variável booleana global e assim que o estado dessa variável assume o valor *true*, o cronômetro é lido. A Figura 23 mostra o trecho do programa com o loop infinito na função *main* e as funções do alarme e do *timer*.

Figura 23 – Trecho do programa 3 modificado para ler médias por tempo.

```

35     while(1) {
36         if (data==true)
37         {
38             value = chrono.ticks();
39             chrono.reset();
40             chrono.start();
41             cout << value << " " << count << endl;
42             count = 0;
43             data = false;
44         }
45     }
46 }
47
48 void alarm_func(void)
49 {
50     data = true;
51 }
52
53 void timer_func(void)
54 {
55     count ++;
56 }

```

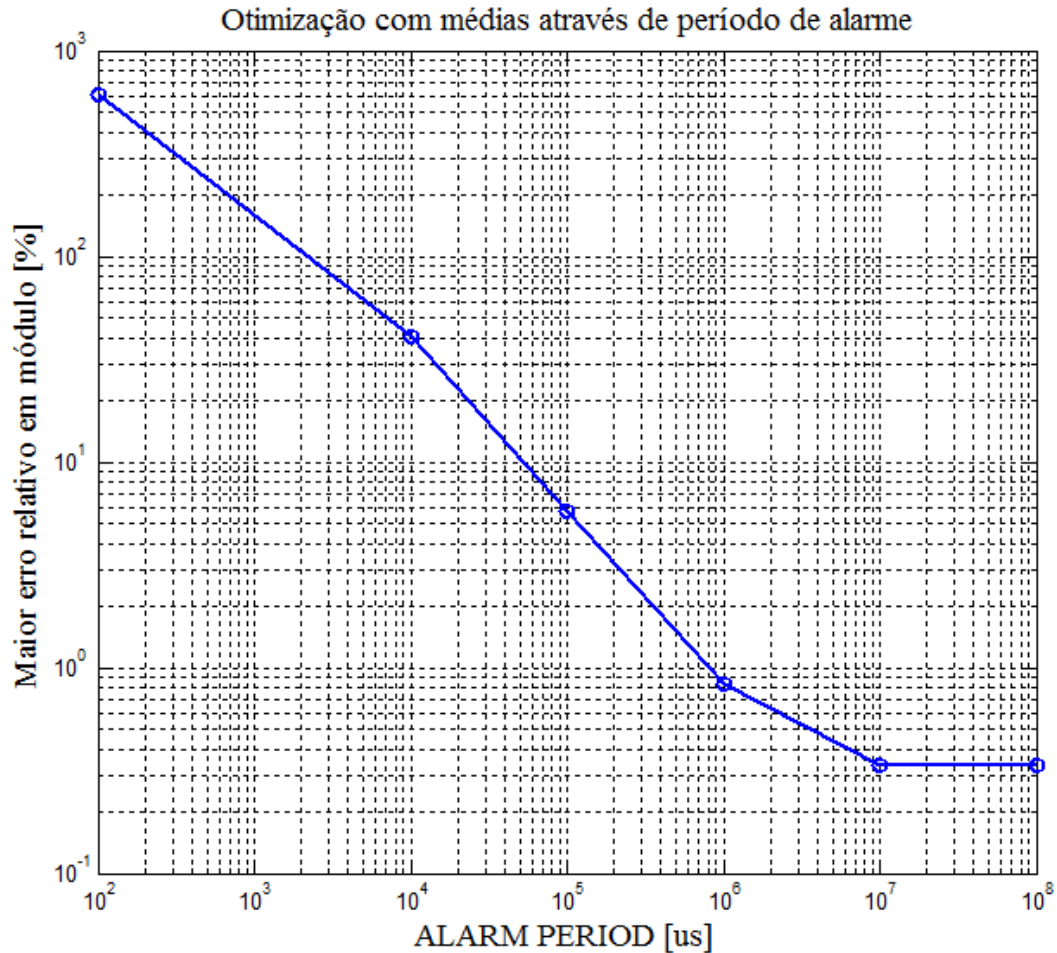
Fonte: autor.

Para esse método, a frequência dos pulsos lidos (f) é dada pela seguinte expressão,

$$f = \text{count} \cdot \text{comp1} \cdot \frac{10000}{\text{value}} \quad (13)$$

Embora não apareça na expressão acima para a frequência, o parâmetro a ser otimizado agora é o período do alarme. A fim de se comparar esse método com o método de médias através de número fixo de amostras, variou-se o período do alarme com a frequência mantida em 14750Hz e com comp1 mantido em 104. O resultado dessa otimização é mostrado abaixo na Figura 24.

Figura 24 – Otimização do programa 3 – Médias através de período de alarme.



Fonte: autor.

Para um período de alarme de 1s o erro relativo obtido foi 0,827% e, portanto, 0,101% maior que na condição original. Já para um período de 100ms o erro foi de 5,763%. Apenas com o período do alarme configurado em 10s e também em 100s o erro diminuiu para 0,334%, porém esses períodos de amostragem são impraticáveis. A Figura 24 reforça a suposição da existência de uma assíntota em torno de 0,33%.

Após os resultados apresentados pelas Figuras 22 e 24, constata-se que o método de médias através de número fixo de amostras é o mais efetivo para reduzir o erro relativo e, por isso, foi o método adotado. Além do mais, esse método garante que o valor do cronômetro é lido imediatamente após o incremento do contador global. No outro método, o valor do cronômetro pode ser lido entre um incremento e outro, o que gera a perda de alguns pulsos no resultado da leitura.

3.5.2. Envio de mensagens

Uma vez que o grande diferencial deste trabalho é possibilitar uma medição sem fio da vibração torsional, mensagens devem ser enviadas pelos EPOSMoteII conectados aos sensores para um EPOSMoteII conectado a um computador localizado no *cockpit* do veículo. Essas mensagens devem conter a identificação da placa que está enviando e sua respectiva faixa de operação e obviamente o valor lido pelo cronômetro. Para a identificação da placa que está enviando a mensagem e da sua respectiva faixa de operação foi criada uma variável do tipo *unsigned short*, que tem o tamanho de 2 bytes. Desse modo, quando essa variável assume o valor “11” significa que a placa 1 está enviando a mensagem e sua faixa de operação é a 1. De maneira semelhante, quando essa variável assume o valor “23” significa que a placa 2 está enviando a mensagem e sua faixa de operação é a 3, e assim por diante. Já para enviar o valor lido pelo cronômetro, foi utilizada uma variável do tipo *volatile unsigned long long TSC::Time_Stamp*, visto que esse é o tipo de variável definido na árvore do EPOS para receber o valor do cronômetro. O trecho do programa que define a estrutura da mensagem enviada é apresentado a seguir na Figura 25.

Figura 25 – Trecho do programa que define a estrutura da mensagem enviada.

```

13 struct Message
14 {
15     unsigned short id_send;
16     volatile TSC::Time_Stamp tx_value;
17 } Msg;

```

Fonte: autor.

Para os EPOSMoteII conectados aos sensores foi criada uma função para envio de mensagens. Essa função recebe como parâmetros justamente a identificação da placa e de sua faixa de operação e o valor lido pelo cronômetro e os associa com a estrutura da mensagem mostrada anteriormente. Então, uma mensagem é enviada através da Network Interface Controller (NIC). O trecho do programa que contém a função para envio de mensagens é representado abaixo pela Figura 26.

Figura 26 – Trecho do programa que contém a função para envio de mensagens.

```

140 void send_msg(unsigned short id, volatile TSC::Time_Stamp value)
141 {
142     Msg.id_send = id;
143     Msg.tx_value = value;
144     int r;
145     while ((r = nic->send(NIC::BROADCAST, (NIC::Protocol) 1, &(Msg), sizeof(Msg))) != 11);
146 }

```

Fonte: autor.

Para o recebimento das mensagens foi criado um programa exclusivo para o EPOSMoteII conectado ao computador no *cockpit* do veículo. Esse deve basicamente receber as mensagens e enviá-las através da UART. Para tanto, foi elaborada uma função chamada *master* que funciona em um loop infinito. Vale a pena ressaltar aqui que para enviar sempre quatro caracteres pela UART, foi criada uma estrutura *if-else*, ou seja, se o valor lido pelo cronômetro for menor que 1000 um 0 é acrescentado na frente desse valor. Isso ocorre pra frequências de pulsos baixas, menores que 1800Hz para a placa 1, por exemplo. O trecho desse programa que contém as funções *master* e *main* é tratado a seguir pela Figura 27. O programa completo para recebimento de mensagens é apresentado no Apêndice A.

Figura 27 – Trecho do programa para recebimento de mensagens.

```

22 void master()
23 {
24     NIC::Protocol prot;
25     NIC::Address src;
26
27     while(true)
28     {
29         while(!(nic->receive(&src, &prot, &(Msg), sizeof(Msg)) > 0));
30         if(Msg.tx_value < 1000)
31         {
32             cout << Msg.id_send << "0" << Msg.tx_value << endl;
33         }
34         else cout << Msg.id_send << Msg.tx_value << endl;
35     }
36 }
37
38 int main()
39 {
40     nic = new NIC();
41     master();
42 }

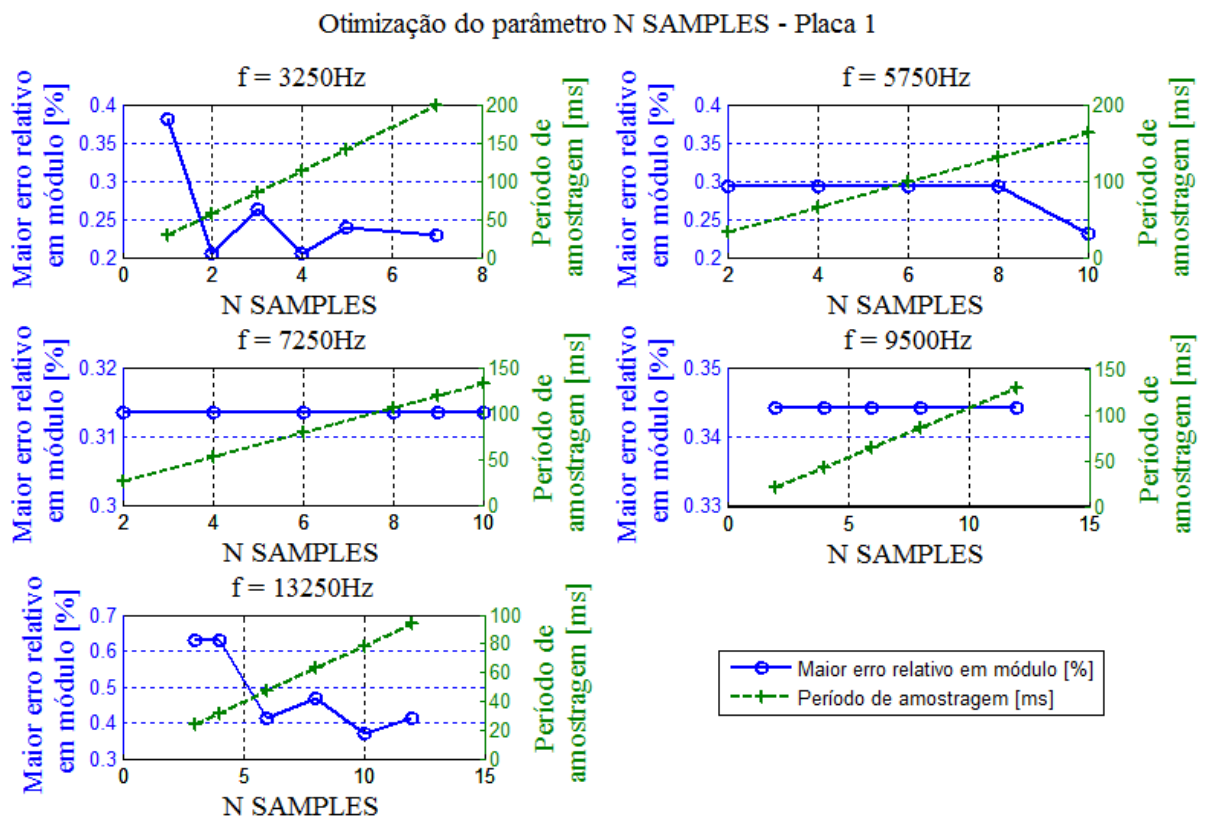
```

Fonte: autor.

3.5.3. Programa final

Uma vez escritos os programas separadamente para leitura do sensor magnético e para envio de mensagens é necessário integrá-los em um único programa. Conforme descrito anteriormente, a otimização do programa 3 com médias através de número fixo de amostras mostrou-se efetiva na redução do erro relativo de leitura. Como todos os testes e otimizações anteriores foram feitos sem levar em conta o envio de mensagens, a otimização do parâmetro `N_SAMPLES` foi feita para cada placa levando a comunicação sem fio em consideração. Desse modo, otimizou-se o parâmetro `N_SAMPLES` para cada uma das faixas de operação obtidas anteriormente. Durante essa otimização, o período de amostragem também foi avaliado. O resultado obtido para a placa 1 é apresentado abaixo na Figura 28.

Figura 28 – Otimização do parâmetro `N_SAMPLES` para a placa 1.



Fonte: autor.

Como era esperado, o período de amostragem é diretamente proporcional ao número de amostras. Além disso, existe um compromisso entre o erro relativo e o período de amostragem, ou seja, eles são conflitantes. Para a escolha do parâmetro `N_SAMPLES`, adotou-se o seguinte critério: o período de amostragem deve ser inferior a 50ms para as

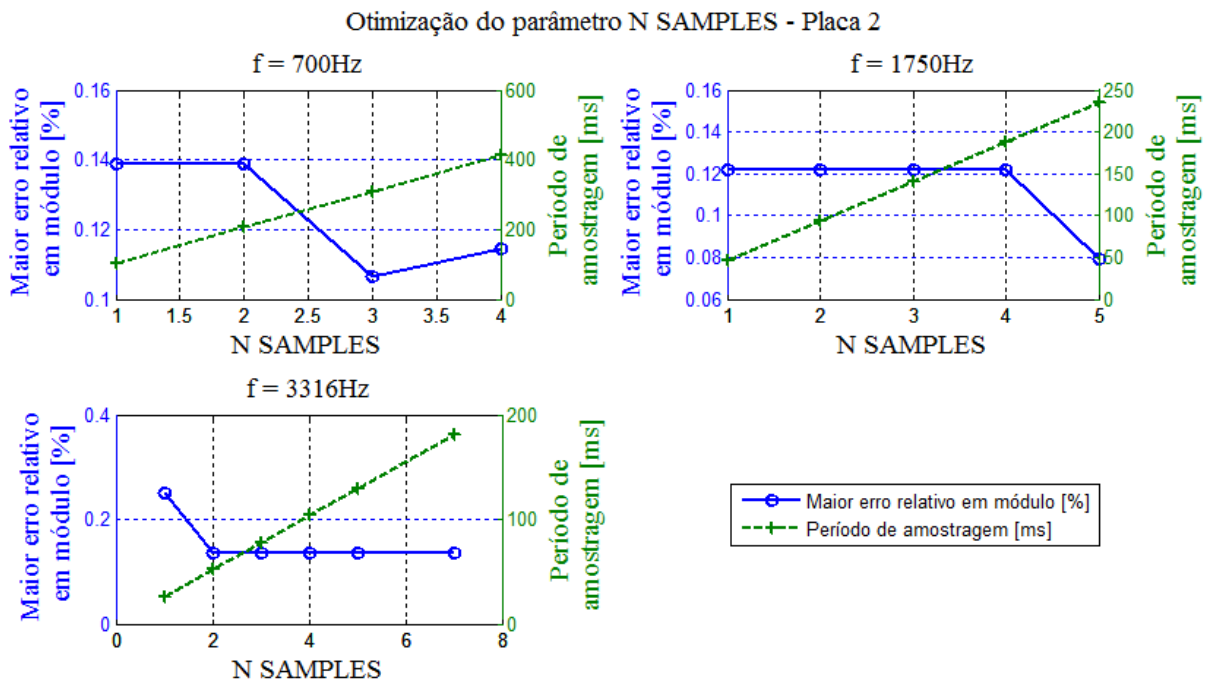
frequências de pulsos avaliadas aqui e o erro relativo deve ser o menor possível. Assim, o parâmetro N_SAMPLES foi definido para cada faixa de operação, como mostrado pela Tabela 3.

Tabela 3 – Definição do parâmetro N_SAMPLES para cada faixa de operação da placa 1.

Faixa de operação	Frequência [Hz]	comp1	N_SAMPLES
1	$f < 5000$	92	2
2	$5000 \leq f < 6500$	94	2
3	$6500 \leq f < 8000$	96	2
4	$8000 \leq f < 11000$	102	2
5	$f \geq 11000$	104	6

De maneira semelhante ao feito para a placa 1, o parâmetro número de amostras também foi otimizado para a placa 2 e o resultado disso é descrito a seguir pela Figura 29.

Figura 29 – Otimização do parâmetro N_SAMPLES para a placa 2.



Fonte: autor.

Como a placa 2 trabalha com frequências mais baixas que a placa 1, os períodos de amostragem obtidos foram maiores nesse caso. Adotou-se o seguinte critério para a seleção do parâmetro N_SAMPLES: o período de amostragem deve ser inferior a 100ms e o erro

relativo deve ser o menor possível. Dessa forma, o parâmetro N_SAMPLES foi definido para cada faixa de operação, como mostrado pela Tabela 4.

Tabela 4 – Definição do parâmetro N_SAMPLES para cada faixa de operação da placa 2.

Faixa de operação	Frequência [Hz]	comp1	N_SAMPLES
1	$f < 1000$	72	1
2	$1000 \leq f < 2500$	82	1
3	$f \geq 2500$	86	2

Um resultado importante obtido das otimizações do parâmetro N_SAMPLES foi o período de amostragem mínimo suportado por essa comunicação sem fio. Constatou-se que para a estrutura de mensagem considerada aqui ocorrem erros de comunicação para períodos de envio de mensagens menores que cerca de 17,5ms. Consequentemente, pode-se afirmar que o limite máximo para a frequência de amostragem do EPOSMoteII funcionando com os programas descritos anteriormente é aproximadamente 57Hz.

Com os parâmetros comp1 e N_SAMPLES otimizados para todas as faixas de operação das duas placas, deve-se criar uma lógica para mudança de faixa. Para tanto foi empregada uma estrutura do tipo *switch-case* que avalia o valor de uma variável denominada *range*. O valor dessa variável define a faixa de operação e as mudanças de faixas ocorrem quando o valor lido pelo cronômetro atinge certos limites obtidos empiricamente. Um trecho do programa final para a placa 1 que mostra o uso dessa estrutura para mudança de faixa é mostrado pela Figura 30.

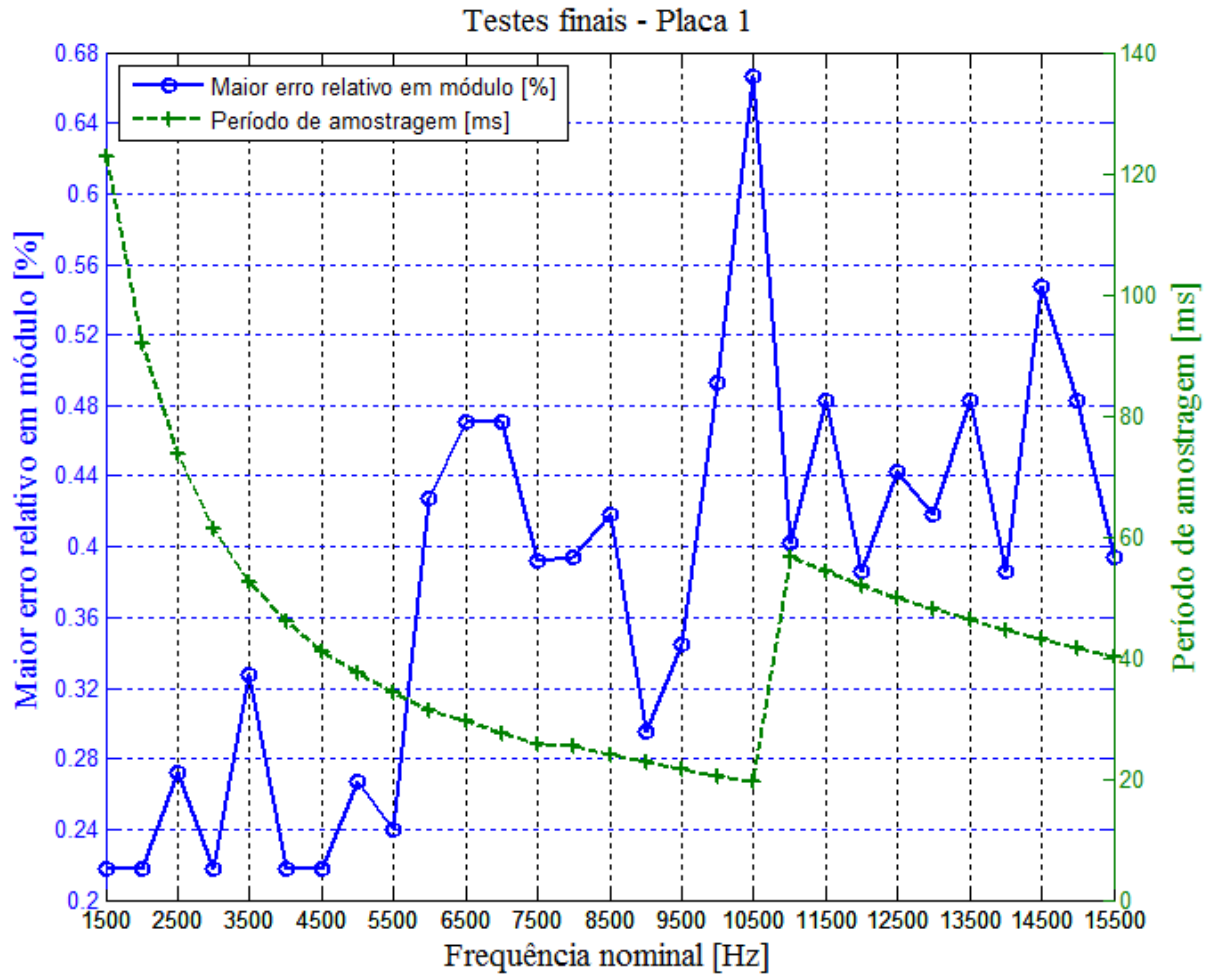
Figura 30 – Trecho do programa final da placa 1 que mostra a estrutura para mudança de faixa.

```
while(1)
{
    if(count==N_SAMPLES)
    {
        switch(range)
        {
            case 1: //Faixa 1 - 1500 a 5000 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(11, sensor_read);
                if(sensor_read < 369) //Critério de mudança pra faixa 2
                {
                    timer.comp1(0x5E); //94 em decimal
                    range = 2; //Muda pra faixa 2
                }
                break;
        }
    }
}
```

Fonte: autor.

Uma vez que a estrutura para mudança de faixa foi estabelecida e os programas para leitura do sensor e para envio de mensagens foram integrados, deve-se testar o sistema completo em toda a faixa de operação de cada placa. O erro relativo de leitura e o período de amostragem são novamente os parâmetros a serem avaliados. Dessa forma, testou-se a placa 1 com pulsos de frequências de 1500Hz a 15500Hz com passos de 500Hz. O resultado desse teste é apresentado pela Figura 31.

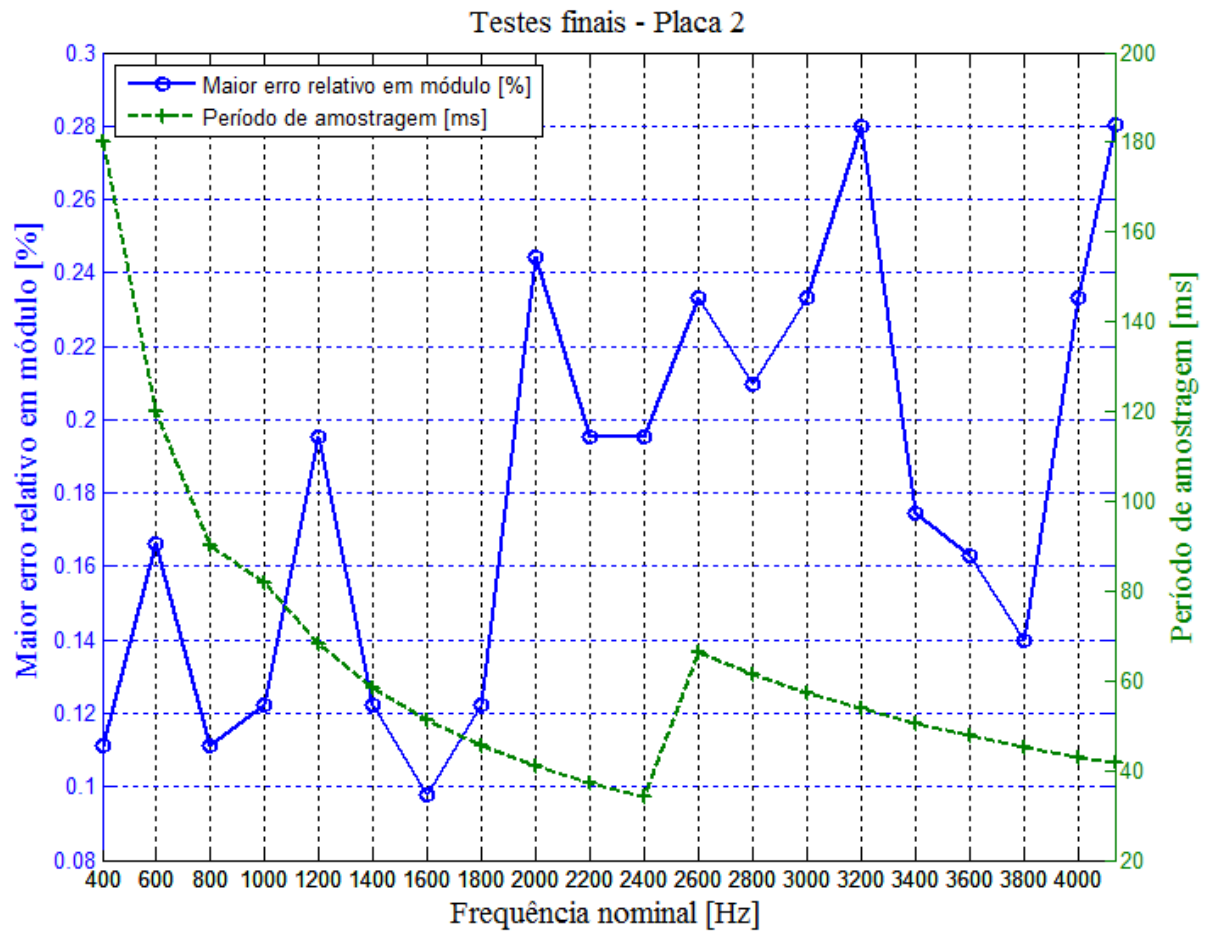
Figura 31 – Teste do programa final para a placa 1.



Fonte: autor.

Observa-se que o período de amostragem é inversamente proporcional à frequência dos pulsos lidos e seu valor variou de cerca de 120ms até 20ms. O salto no ponto de 11000Hz se deve ao número de amostras que aumentou de 2 para 6 na última faixa de operação da placa 1. Por sua vez, o erro relativo de leitura é menor para frequências mais baixas e oscilou de 0,21% a 0,66%. Um teste similar foi feito para a placa 2, porém dessa vez com frequências de 400 a 4133Hz com passos de 200Hz. O resultado desse teste é mostrado abaixo pela Figura 32.

Figura 32 – Teste do programa final para a placa 2.



Fonte: autor.

O período de amostragem para a placa 2 assumiu valores de 180ms a 34ms e o salto no ponto de 2600Hz ocorreu novamente em função da mudança do parâmetro N_SAMPLES de 1 para 2 na última faixa de operação. Já o erro relativo foi em geral menor que o obtido para a placa 1 e oscilou de 0,09% a 0,28%. Os programas completos para a leitura do sensor magnético e envio de mensagens pelas placas 1 e 2 são apresentados nos Apêndices B e C, respectivamente.

Cabe mencionar aqui que todos os testes descritos anteriormente para otimização ou validação dos programas para o EPOSMoteII foram realizados com um gerador de sinais modelo 33600A Series da fabricante Keysight. Em todos os testes, aplicou-se um sinal senoidal com amplitude de pico a pico de 300mV diretamente na placa de condicionamento de sinal. A resolução do gerador de sinais vale nesse caso 1 μ Hz.

4. RESULTADOS

4.1. Testes realizados na empresa ZF do Brasil

Com o objetivo de comparar o sistema de medição desenvolvido neste trabalho com o equipamento já utilizado na indústria para a medição da vibração torsional, foram realizados alguns testes na empresa ZF do Brasil. Essa empresa desenvolve, entre outros produtos, embreagens para veículos de passeio e de carga e cabe à equipe de engenharia experimental realizar as medições da vibração torsional durante a fase de homologação das embreagens. Usualmente, mede-se a velocidade de rotação antes e depois da embreagem para avaliar o desempenho do amortecimento torsional oferecido pela mesma.

O veículo instrumentado para os testes foi um Ka Sedan com motor de 3 cilindros, ciclo Otto e câmbio manual de 5 marchas. Um sensor magnético do tipo *pick-up* foi apontado para o volante do motor, de 112 dentes, e o outro sensor foi apontado para a engrenagem no eixo piloto da transmissão, de 45 dentes. Dessa forma, não há uma relação de transmissão entre os dois pontos de medição da velocidade rotacional, ou seja, é possível uma comparação direta da medição dos dois sensores.

Inicialmente, o equipamento de medição usado normalmente nos testes por essa empresa foi conectado aos dois sensores. Trata-se do equipamento *Rotation Analysis System* (RAS) da fabricante Rotec. Esse equipamento consiste em um condicionador de sinal para os sensores, um módulo de aquisição de dados e um *toughbook* para o processamento dos dados. Como o condicionador de sinal e o módulo de aquisição de dados estavam localizados no banco de trás do veículo, foi necessário passar cabos dos sensores localizados sob o capô do veículo para o banco de trás. Em uma primeira tentativa, os cabos foram passados através da porta do motorista de forma que ficavam comprimidos, porém isso gerou muito ruído elétrico e inviabilizou a medição. Após isso, uma segunda tentativa foi feita passando os cabos através da janela do motorista, conforme mostrado pela Figura 33. Dessa vez não houve ruído elétrico gerado pela compressão dos cabos.

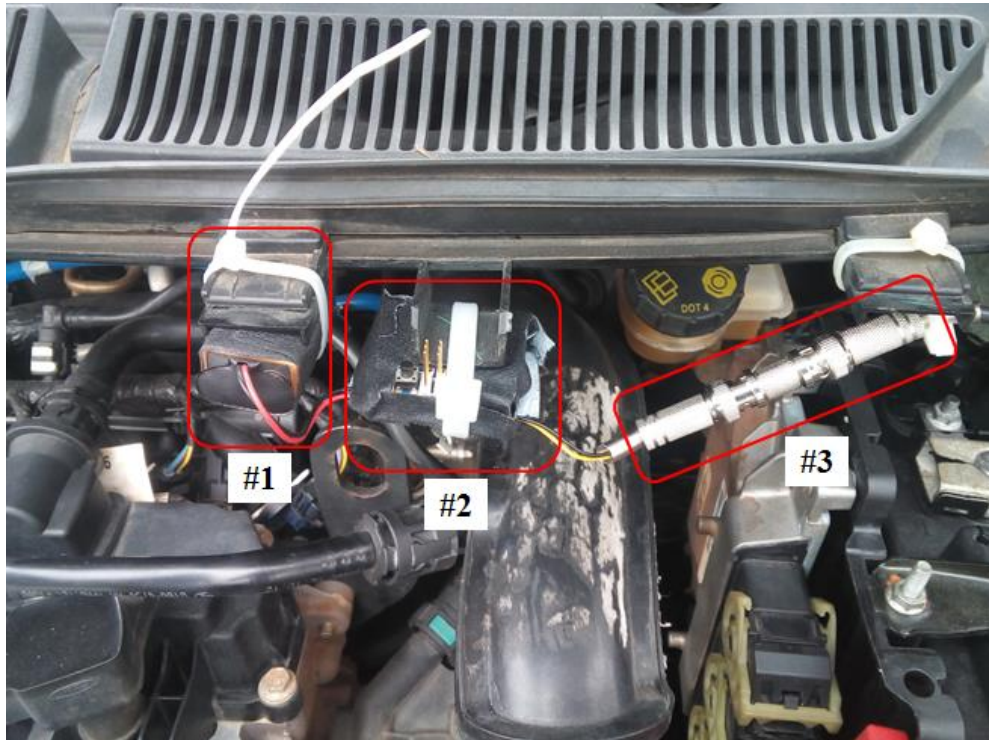
Figura 33 – Passagem de cabos necessária para conectar o Rotec aos sensores.



Fonte: autor.

Ainda em relação à instrumentação, após os testes feitos com o equipamento Rotec, instalou-se o sistema desenvolvido neste trabalho. Em uma primeira tentativa, uma placa foi posicionada sob o capô do veículo, conforme demonstrado pela Figura 34, e a outra placa foi posicionada no banco de trás do veículo utilizando um dos cabos já roteados para a medição com o Rotec. Contudo, a placa posicionada sob o capô do veículo funcionou apenas durante o teste em marcha lenta e parcialmente durante o primeiro teste em tração, após isso nenhum sinal foi recebido por essa placa. Conclui-se disso que a temperatura sob o capô do veículo era muito alta e impedia o correto funcionamento dos circuitos. Em uma segunda tentativa, as duas placas foram posicionadas no banco de trás do veículo utilizando os cabos já passados anteriormente para conectar os sensores e dessa vez foi possível realizar todos os testes em tração sem perda de sinal.

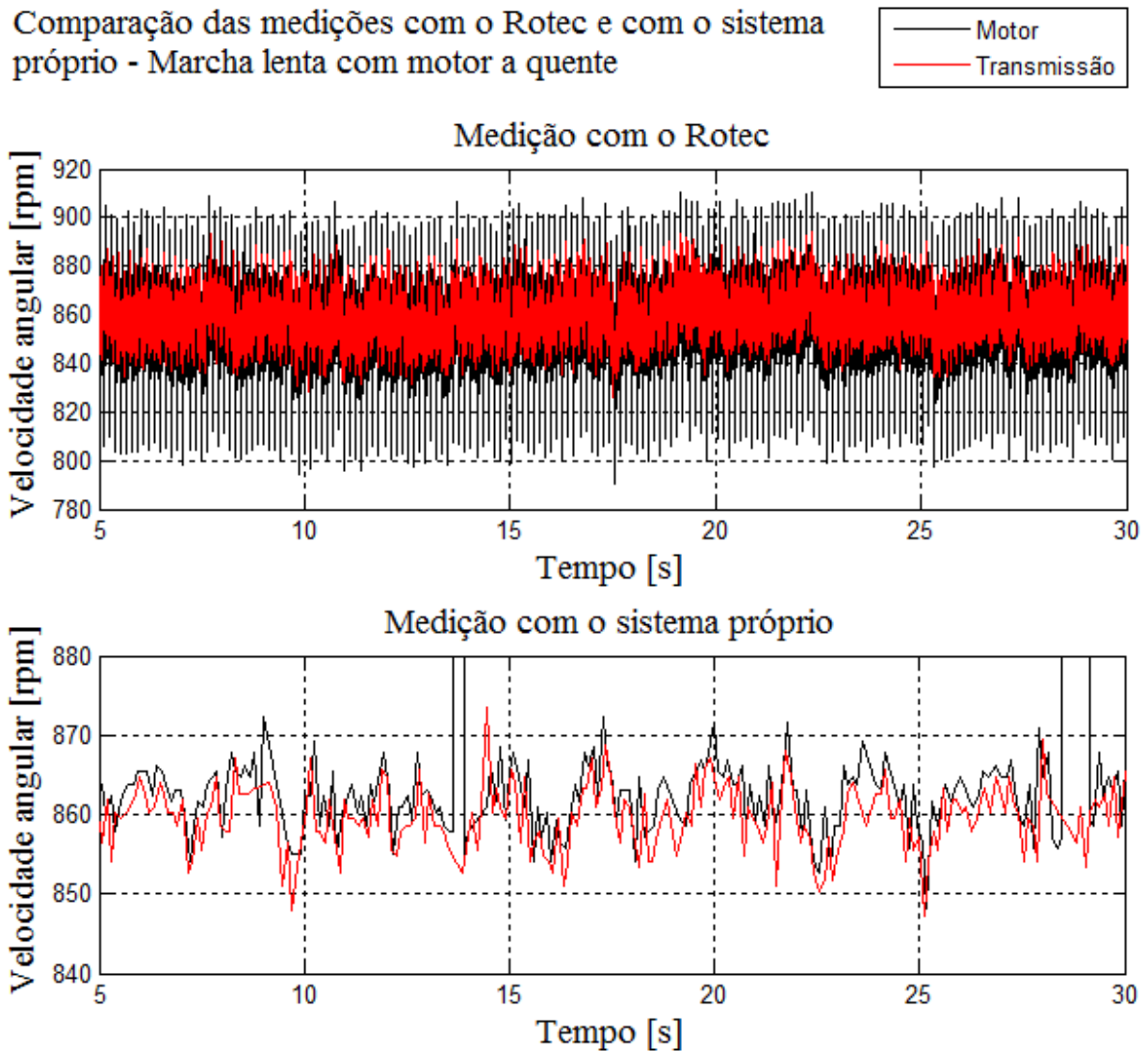
Figura 34 – Sistema próprio posicionado sob o capô do veículo; #1 – Bateria de 9V; #2 – Placa de condicionamento de sinal mais EPOSMoteII; #3 – Conectores BNC para o sensor.



Fonte: autor.

Em relação aos testes em si, primeiro eles foram feitos com o equipamento Rotec e depois foram feitos com o sistema próprio desenvolvido neste trabalho. Foram realizados dois tipos de teste diferentes: em marcha lenta e de tração. O teste em marcha lenta foi feito com o veículo parado e com o câmbio na posição de neutro. Durante esse tipo de teste não se acelera o veículo. O teste em marcha lenta foi realizado com o motor a frio e com o motor a quente, mas como os resultados obtidos foram muito semelhantes apenas o teste com o motor a quente é mostrado aqui. Por sua vez, os testes de tração foram feitos com o veículo em pista. Com uma marcha específica engatada, acelerou-se o veículo da rotação de marcha lenta até próximo da rotação de corte do motor, que para o veículo testado vale cerca de 7000rpm. É importante destacar que não houve troca de marchas durante os testes de tração. Os testes foram feitos progressivamente com a 2ª, 3ª, 4ª e 5ª marcha engatadas. Uma vez que os resultados obtidos pelos testes de tração não mudaram significativamente em função da marcha, apenas os resultados para a 2ª e a 3ª marcha são apresentados aqui. O resultado do teste em marcha lenta com o motor a quente é mostrado na Figura 35.

Figura 35 – Resultado do teste em marcha lenta com o motor quente realizado na empresa ZF do Brasil.



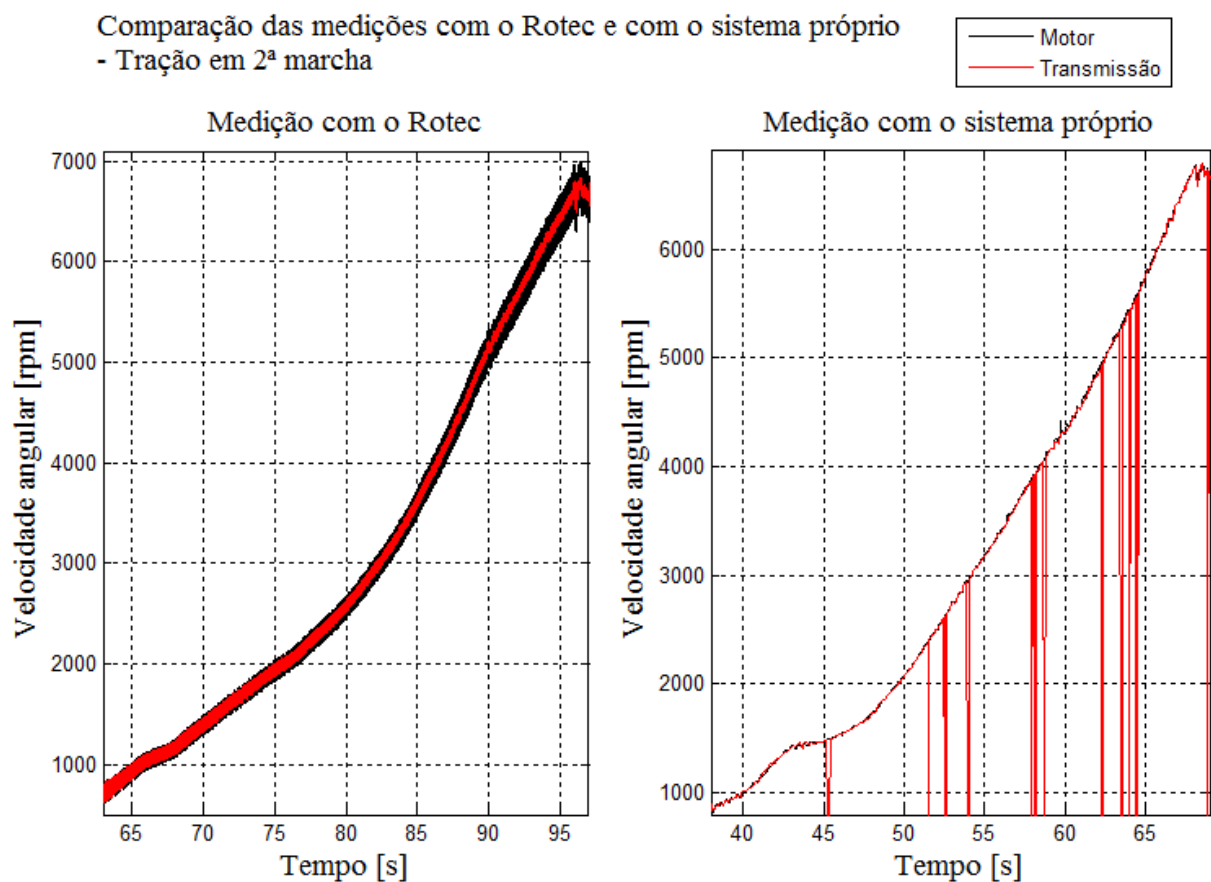
Fonte: autor.

Observa-se que a taxa de amostragem do Rotec é muito maior que a taxa de amostragem do sistema próprio. De fato, o período de amostragem do Rotec para o motor durante esse teste em marcha lenta foi em média 0,6ms, ao passo que o período médio de amostragem do sistema próprio foi 114ms. Uma consequência importante disso é a inviabilização da análise de ordem da vibração a partir da medição com o sistema próprio. Além do mais, na medição com o Rotec é possível distinguir a diferença entre a amplitude da oscilação da velocidade angular do motor e da transmissão, e, portanto, o amortecimento torsional oferecido pela embreagem, o que não é notado na medição com o sistema próprio. A velocidade angular do motor, por exemplo, oscilou de 800rpm a 900rpm com o Rotec e

apenas de 850rpm a 870rpm com o sistema próprio. Uma característica comum às duas medições foi a velocidade média tanto do motor quanto da transmissão de cerca de 860rpm.

Erros de leitura significativos foram observados na medição com o sistema próprio. Na Figura 35, nota-se que a velocidade angular assume valores extremamente elevados nos instantes de 13s e 28s. Esses erros ocorrem somente quando os dois módulos de comunicação enviam mensagens ao mesmo tempo. O resultado do teste de tração com a 2ª marcha engatada é apresentado em seguida pela Figura 36.

Figura 36 – Resultado do teste de tração em 2ª marcha realizado na empresa ZF do Brasil.

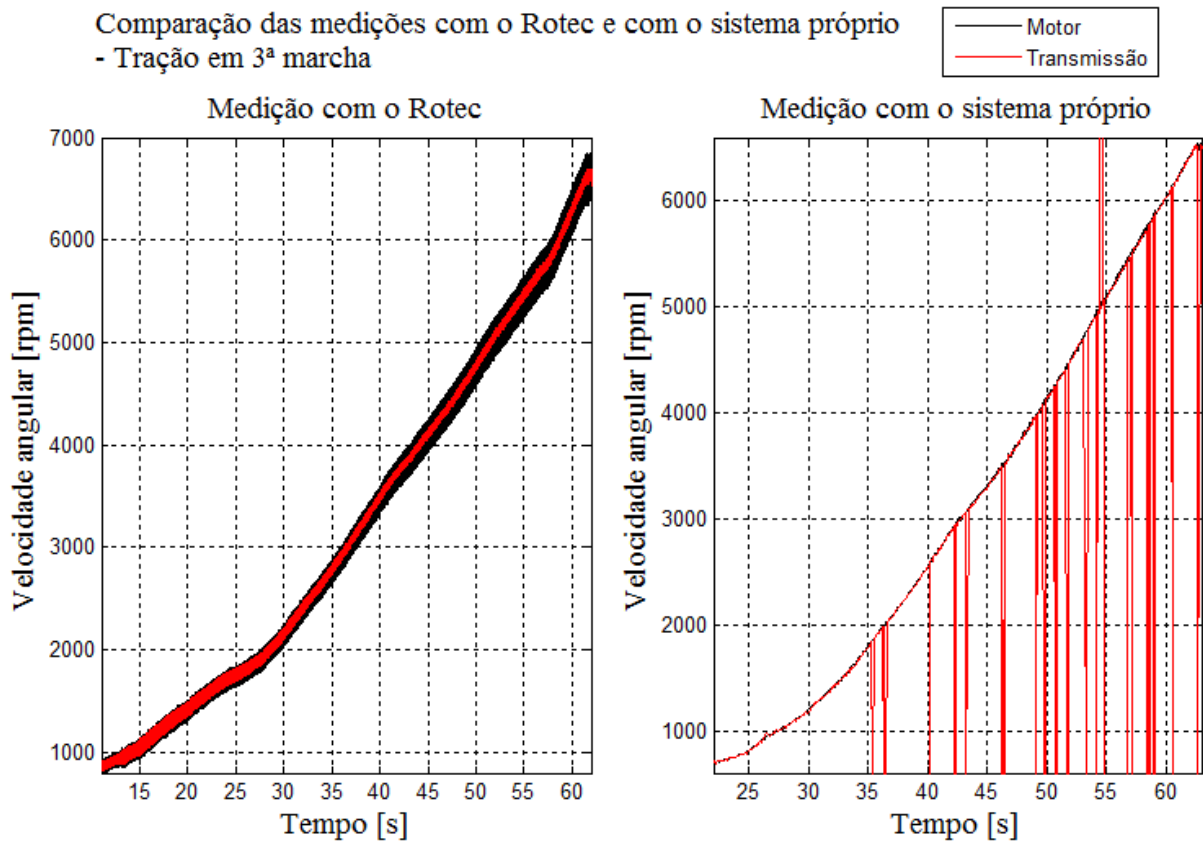


Fonte: autor.

Como se trata de um teste durante o qual o veículo é ininterruptamente acelerado, o formato da curva obtida é muito semelhante entre as duas medições, embora elas não tenham sido realizadas simultaneamente. Em ambos os casos, a velocidade cresceu aproximadamente de 800rpm a 6700rpm. A diferença marcante entre as duas medições está na amplitude de oscilação das velocidades. Novamente, é notado que a taxa de amostragem do Rotec é muito maior que a do sistema próprio. Erros de leitura significativos ocorreram na medição com o

sistema próprio e são representados pelos traços verticais no gráfico. Ao contrário do que ocorreu no teste em marcha lenta, os erros de leitura significativos aqui correspondem a valores lidos como zero. A seguir, o teste de tração em 3ª marcha é retratado pela Figura 37.

Figura 37 – Resultado do teste de tração em 3ª marcha realizado na empresa ZF do Brasil.



Fonte: autor.

Todas as observações feitas para o teste de tração em 2ª marcha se aplicam para o teste em 3ª marcha. A diferença é que nesse caso foram lidos valores tanto extremamente altos como zeros. Como explicado anteriormente, os resultados dos testes de tração em 4ª e 5ª marchas foram muito semelhantes aos já mostrados aqui e, por isso, não serão tratados.

Tecnicamente, o sistema de medição da fabricante Rotec é muito superior ao desenvolvido neste trabalho. Com os canais digitais, o intervalo de tempo entre as bordas de subida ou de descida é medido através de um *timer* de 40bits que opera sobre um *clock* de 10GHz. Dessa forma, o Rotec é capaz de medir o tempo entre a passagem de cada dente da engrenagem. Por sua vez, o módulo de comunicação EPOSMoteII opera com um oscilador externo de cristal de 24MHz e seus *timers* possuem 16bits. A frequência do cronômetro usado pelo EPOSMoteII para a medição do tempo entre certa quantidade de dentes vale 10kHz.

Uma engrenagem de 112 dentes girando a 6000rpm equivale a um trem de pulsos com frequência de 11,2kHz, e, portanto, um cronômetro de no mínimo 22,4kHz de frequência deveria ser usado para medir o tempo entre os pulsos. Em contrapartida, o equipamento RAS como um todo custa cerca de R\$220.000,00, enquanto o sistema de medição desenvolvido neste trabalho é estimado em R\$300,00 (esse valor se refere apenas ao custo de 3 módulos de comunicação EPOSMotII).

Existem três fontes principais de erro na medição da velocidade angular com a metodologia usada durante esses testes. A primeira fonte de erro é a própria medição dos períodos do trem de pulsos. Uma grande quantidade de *clocks* por período de pulso permite a medição dos períodos do trem de pulsos com uma exatidão de ± 1 *clock*, ou seja, ocorre apenas um erro de quantização. Novamente, para uma engrenagem de 112 dentes girando a 6000rpm, a frequência dos pulsos gerados vale 11,2kHz. Como o *clock* do Rotec trabalha a 10GHz, a divisão desse valor do *clock* pela frequência dos pulsos gerados corresponde a $8,9 \times 10^5$ incrementos do *clock* por pulso, o que representa um erro relativo de $1,1 \times 10^{-4}\%$. A segunda fonte de erros é a vibração do sensor magnético ou o movimento relativo entre o sensor e o seu alvo e a terceira fonte de erros é a variação no espaçamento entre os dentes da engrenagem (ADAMSON, 2004). As duas últimas fontes de erro relatadas não foram quantizadas neste trabalho. Dessa forma, assume-se que o erro relativo máximo do Rotec durante esses testes foi de $1,1 \times 10^{-4}\%$. Já o erro relativo máximo do sistema próprio foi considerado como 0,66% de acordo com a Figura 31. Um comparativo entre o Rotec e o sistema próprio é dado pela Tabela 5.

Tabela 5 – Comparativo entre o Rotec e o sistema próprio.

Grandeza	Rotec	Sistema próprio
Frequência do <i>clock</i>	10GHz	24MHz
<i>Timer</i>	40bits	16bits
Maior erro relativo apresentado durante os testes	$1,1 \times 10^{-4}\%$	0,66%
Custo	R\$220.000,00	R\$300,00

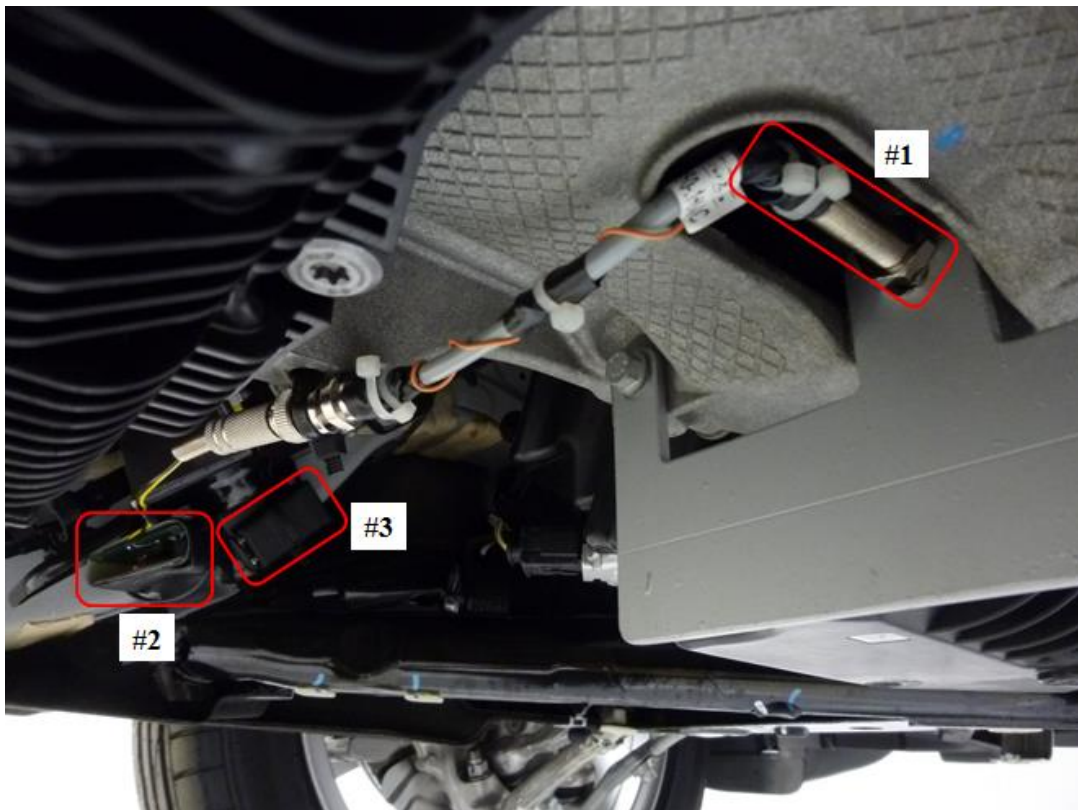
4.2. Testes realizados na empresa BMW do Brasil

Ainda com o objetivo de validar o sistema de medição desenvolvido neste trabalho foram realizados alguns testes na empresa BMW do Brasil dentro da área do time de análise de causa raiz, parte integrante do departamento de qualidade do veículo completo. De forma

diferente dos testes feitos na empresa ZF do Brasil, as medições com o sistema próprio e com o equipamento usado para comparação foram simultâneas, o que permitiu uma comparação direta dos dois sinais no tempo.

O veículo instrumentado foi um BMW 320i com motor de 4 cilindros, ciclo Otto e câmbio automático de 8 marchas. Como foi usado um veículo de produção, não foi possível instalar um sensor na caixa de transmissão, pois seria necessário criar um furo na mesma. Dessa forma, apenas um sensor magnético foi posicionado no volante do motor, que contém 115 dentes. A instrumentação realizada com o sistema próprio é mostrada a seguir pela Figura 38.

Figura 38 – Instrumentação com o sistema próprio; #1 – Sensor magnético; #2 – Placa de comunicação EPOSMoteII mais placa de condicionamento; #3 – Bateria de 9V.

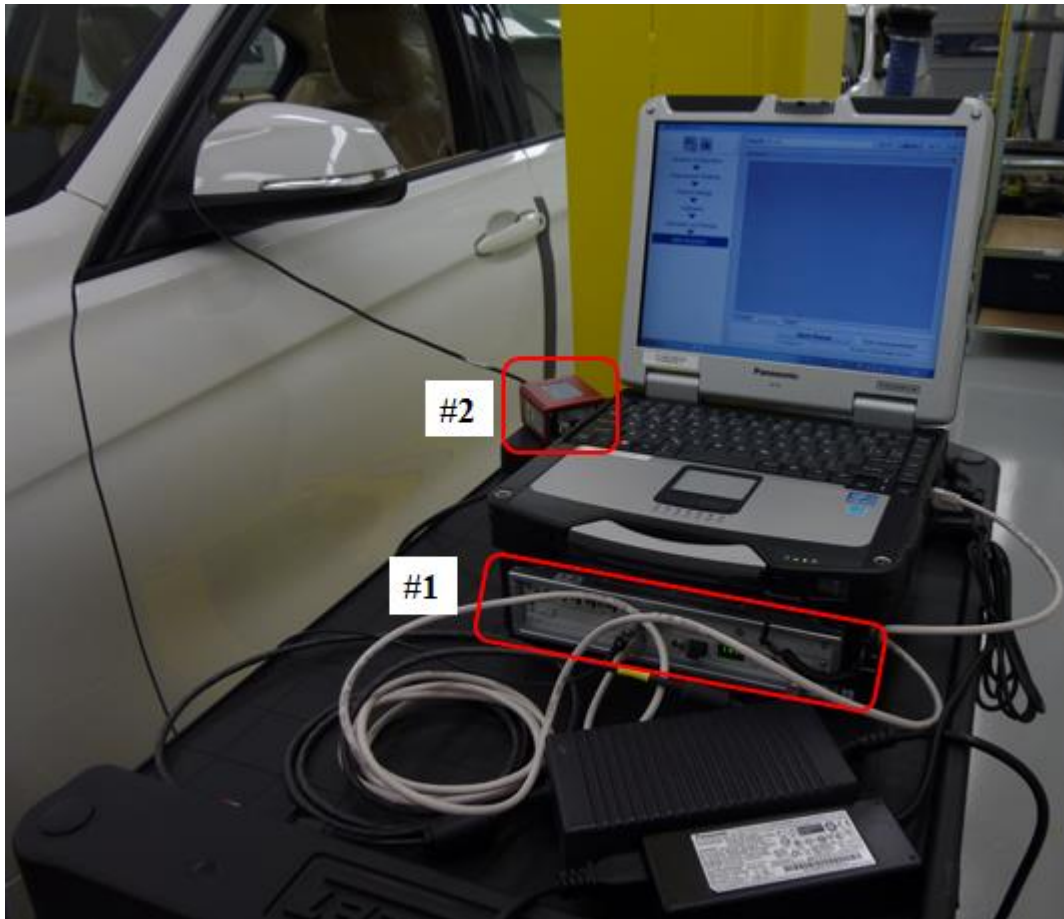


Fonte: autor.

O equipamento usado para validação foi o PAK MKII da fabricante Müller-BBM. Esse equipamento é usado para a medição precisa e de alta velocidade de sinais analógicos e digitais e possui *software* próprio para pós-processamento dos dados. Para alimentar o canal digital do PAK MKII foi usado um conversor de sinais da fabricante TBJ-Messtechnik. Esse conversor de sinais fornece um sinal TTL através da leitura da rotação do motor via linha

CAN e é conectado ao veículo por um cabo OBD2. A instrumentação realizada com o sistema da Müller-BBM é ilustrada pela Figura 39.

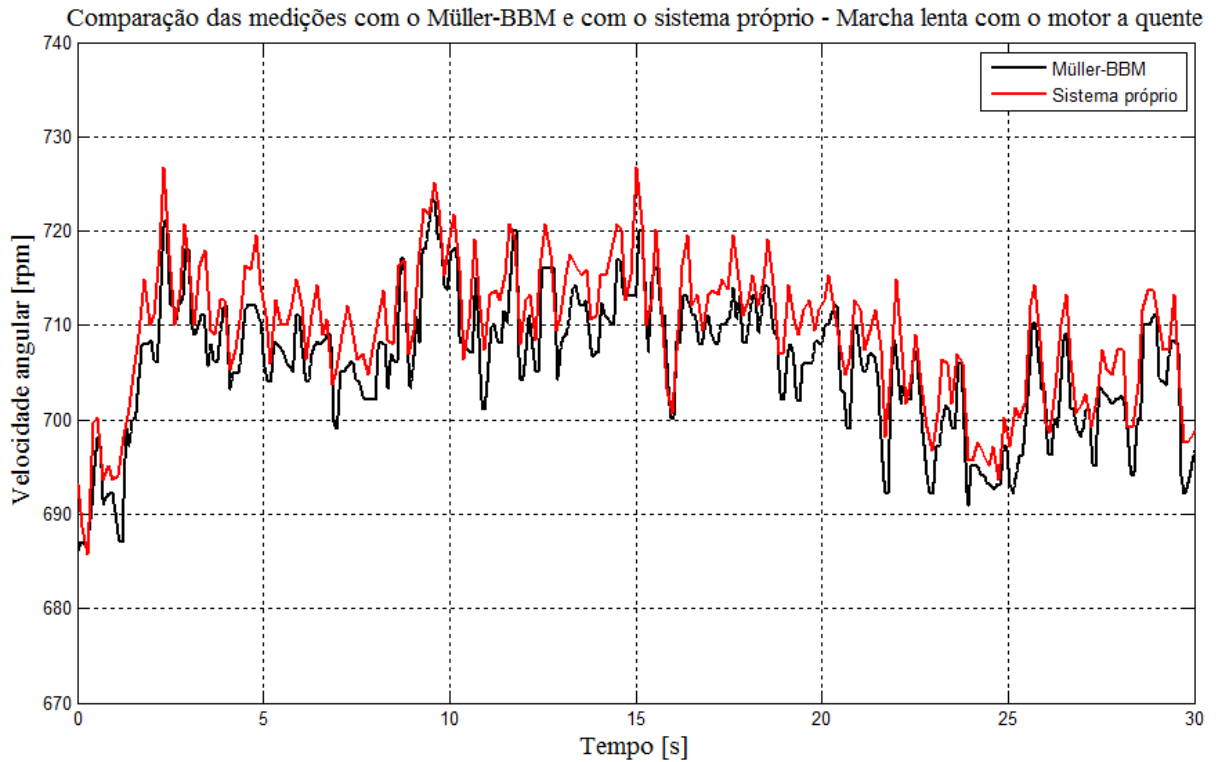
Figura 39 – Instrumentação com o sistema da fabricante Müller-BBM; #1 – PAK MKII; #2 – Conversor de sinais.



Fonte: autor.

Todos os testes foram feitos em um ambiente fechado com o veículo sobre um elevador. Testes em pista poderiam acumular algum tipo de sujeira no volante do motor e não foram viáveis por se tratar de um veículo de cliente. Três tipos de teste diferentes foram executados: em marcha lenta, de tração em 2ª marcha e com troca de marchas da 1ª até a 8ª marcha. O resultado para o teste em marcha lenta é apresentado em sequência pela Figura 40.

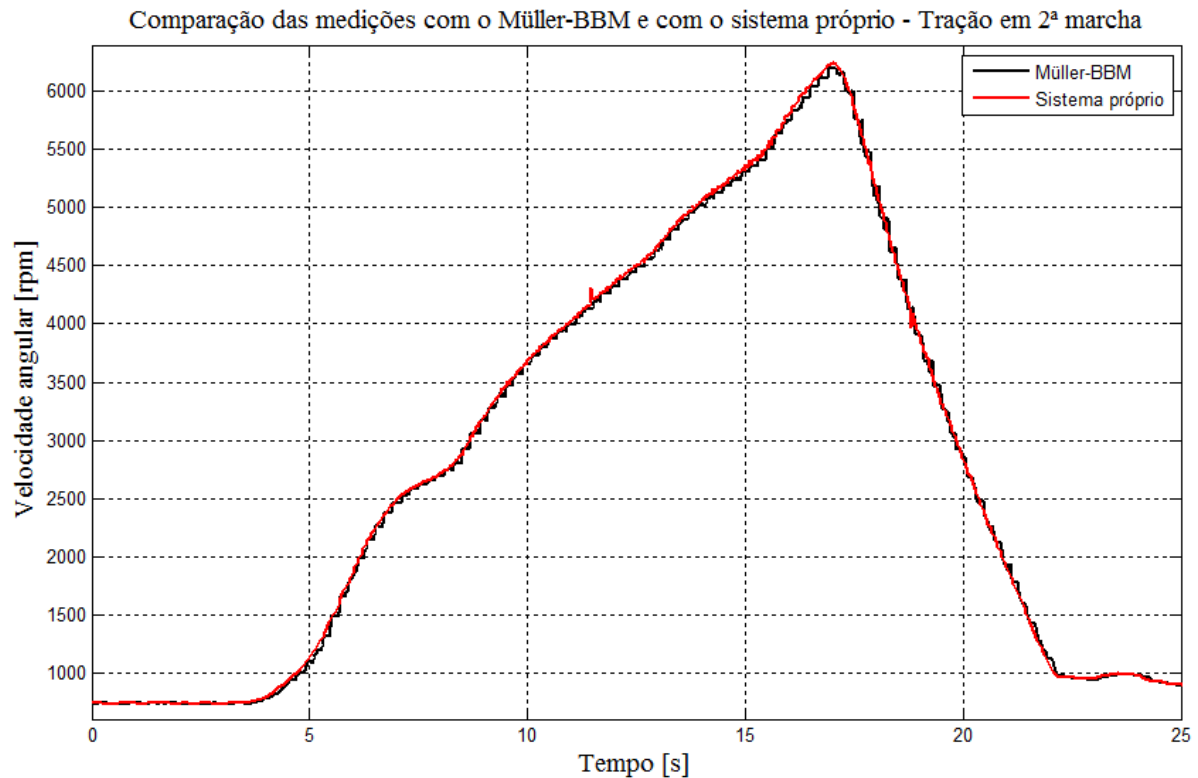
Figura 40 – Resultado do teste em marcha lenta realizado na empresa BMW do Brasil.



Fonte: autor.

Observa-se que as duas curvas apresentaram formatos muito semelhantes e oscilaram em torno dos 700rpm, porém houve um offset entre elas. A diferença média calculada entre as duas medições foi de 13rpm. Nota-se também que a medição com o sistema próprio apresentou picos mais bem definidos do que a medição com o equipamento Müller-BBM, que apresentou picos mais arredondados e de amplitude menor. Ao contrário dos testes feitos na empresa ZF do Brasil, erros de leitura significativos não ocorreram aqui, já que apenas um sensor foi utilizado. O resultado do teste de tração em 2ª marcha é mostrado pela Figura 41.

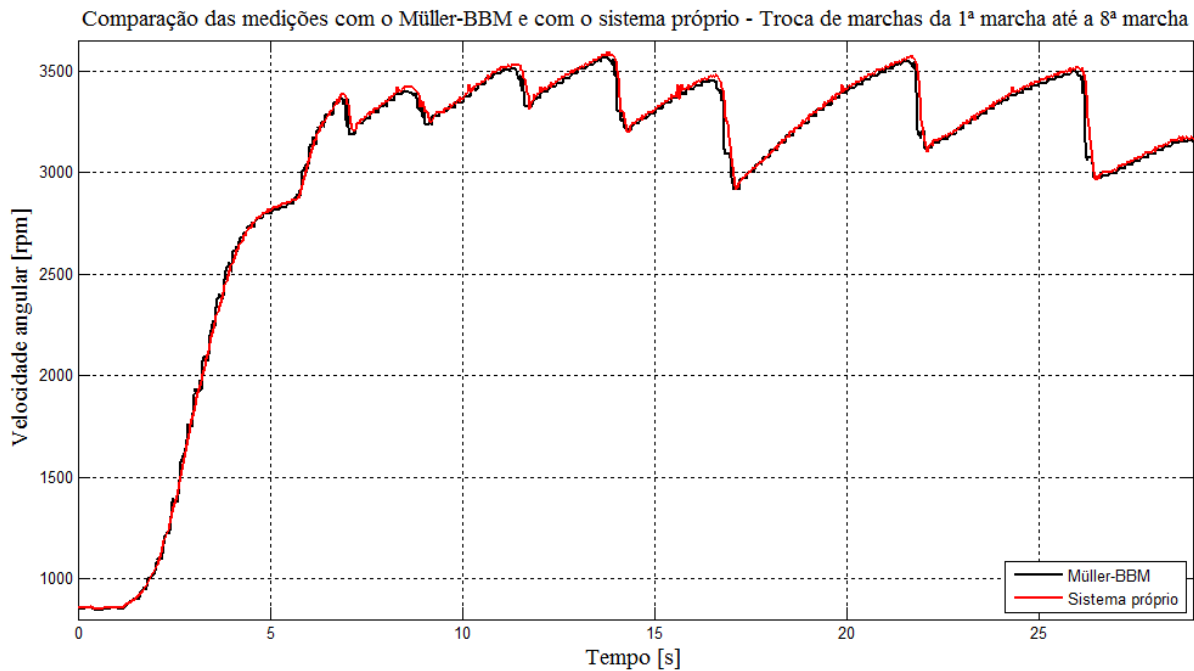
Figura 41 – Resultado do teste de tração em 2ª marcha realizado na empresa BMW do Brasil.



Fonte: autor.

Durante esse teste, acelerou-se o veículo da rotação em marcha lenta até próximo da rotação de corte do motor e em seguida o acelerador foi liberado, o que fez a rotação do motor reduzir até a de marcha lenta novamente. A velocidade angular variou aproximadamente de 740rpm a 6200rpm. Nota-se que a medição com o sistema próprio acompanhou de forma bem próxima a medição com o equipamento Müller-BBM e não houve erro de leitura significativo. É possível notar também que a medição com o equipamento Müller-BBM apresentou degraus ao longo de toda a faixa de rotação medida. Em outras palavras, a discretização com o sistema próprio foi melhor que a discretização com o equipamento de validação. A razão para isso está no número de pulsos gerados a cada rotação do motor. Como o sensor magnético do sistema próprio foi apontado para o volante do motor, 115 pulsos foram gerados por revolução. Por sua vez, o canal digital do Müller-BBM foi alimentado por um conversor de sinais conectado à linha CAN do veículo e esse conversor de sinais gera apenas 1 pulso por revolução do motor. O último teste realizado foi o de troca de marchas da 1ª até a 8ª marcha e o seu resultado é apresentado a seguir pela Figura 42.

Figura 42 – Resultado do teste com troca de marchas da 1ª até a 8ª marcha realizado na empresa BMW do Brasil.



Fonte: autor.

Como os dois testes anteriores não envolveram variações bruscas da rotação do motor, esse teste foi executado com o objetivo de avaliar a resposta do sistema próprio a variações abruptas do sinal. Assim, acelerou-se o veículo até cerca de 3500rpm e em seguida realizou-se trocas consecutivas de marchas até que a última marcha fosse alcançada. As trocas de marcha são percebidas na Figura 42 pelas quedas repentinas na velocidade angular. O sistema de medição desenvolvido neste trabalho respondeu da mesma forma que o equipamento Müller-BBM às trocas de marcha e nenhum atraso ou erro de leitura significativo foi notado. Aqui também foram observados degraus na medição com o Müller-BBM e isso ocorreu pelo motivo já apontado anteriormente.

Tecnicamente, o conversor de sinais da fabricante TBJ-Messtechnik opera com uma frequência de amostragem de 100Hz e seu erro relativo máximo vale 0,8%. O preço total incluindo todos os *hardwares*, o *software* e a licença para o funcionamento do kit da Müller-BBM é de R\$92.000,00. Por sua vez, conforme demonstrado na seção 3.5.3, o sistema desenvolvido neste trabalho possui uma taxa de amostragem máxima de 57Hz e erro relativo máximo de leitura de 0,66%. Como apresentado na seção anterior, o preço estimado para o sistema desenvolvido aqui é de R\$300,00. A seguir é dado pela Tabela 6 um comparativo entre o sistema próprio e o equipamento de validação.

Tabela 6 – Comparativo entre o Müller-BBM alimentado pelo conversor de sinais e o sistema próprio.

Grandeza	Müller-BBM com conversor de sinais	Sistema próprio
Frequência de amostragem máxima	100Hz	57Hz
Erro relativo máximo	0,8%	0,66%
Custo	R\$92.000,00	R\$300,00

5. CONCLUSÃO

Este trabalho integrou sensores magnéticos comerciais com *hardwares* de comunicação sem fio, apresentou um *software* para comunicação entre os *hardwares*, contemplou o teste do sistema como um todo em veículos e validou os resultados obtidos com equipamentos comerciais. Portanto, os objetivos específicos previamente listados foram cumpridos. Entretanto, o objetivo geral de medir a vibração torsional em um *powertrain* automotivo não foi completamente satisfeito, já que a frequência de amostragem do sistema de medição desenvolvido não foi alta o suficiente para permitir uma posterior análise espectral do sinal.

Durante o desenvolvimento do *software* para a medição da velocidade de rotação, constatou-se que é mais exato medir o tempo entre uma quantidade fixa de pulsos do que medir o número de pulsos entre um período fixo de tempo. Outro resultado importante obtido da programação foi a frequência máxima para o envio de mensagens com o módulo de comunicação EPOSMoteII. Para a estrutura da mensagem considerada neste trabalho este valor foi de aproximadamente 57Hz.

Para a medição da velocidade de rotação no volante do motor e no eixo piloto da transmissão foi necessário construir e programar duas placas diferentes. A primeira placa foi destinada à medição de frequências entre 1500Hz e 15500Hz e apresentou um erro relativo de leitura entre 0,21% e 0,66% com período de amostragem entre 20ms e 120ms. Já a segunda placa foi destinada à medição de frequências entre 400Hz e cerca de 4200Hz e seu erro relativo oscilou de 0,09% a 0,28% com período de amostragem entre 34ms e 180ms. Assim, o erro relativo mostrou-se de forma geral diretamente proporcional à frequência dos pulsos lidos.

O sistema desenvolvido neste trabalho foi testado em duas empresas diferentes. Na empresa ZF do Brasil, a velocidade de rotação foi medida no volante do motor e no eixo piloto da transmissão. Os testes foram feitos primeiro com o equipamento Rotec e em seguida com o sistema próprio. Embora as curvas obtidas com o sistema próprio tenham sido semelhantes às obtidas com o Rotec, erros de leitura significativos ocorreram quando os dois módulos de comunicação estavam transmitindo mensagens ao mesmo tempo. Além disso, não foi possível realizar uma análise da vibração torsional a partir das medições com o sistema próprio em função da baixa frequência de amostragem.

Por sua vez, na empresa BMW do Brasil, a velocidade de rotação do motor foi medida simultaneamente pelo sistema próprio e pelo equipamento Müller-BBM alimentado por um conversor de sinais. As curvas obtidas das medições com o sistema próprio acompanharam de forma muito próxima as curvas provenientes das medições com o Müller-BBM e a discretização da velocidade angular foi até mesmo melhor no sistema próprio. Além do mais, o sistema próprio apresentou boa resposta dinâmica no teste com trocas de marcha.

O sistema desenvolvido aqui pode ser uma alternativa de baixo custo para os equipamentos já existentes no mercado, porém para a sua comercialização e aplicação na indústria dois aspectos precisam ser aperfeiçoados: a eliminação dos erros de leitura significativos quando os dois módulos de comunicação enviam mensagens ao mesmo tempo e a taxa de amostragem. Para a avaliação da vibração torsional, deve-se idealmente medir o período de cada dente da engrenagem e não a média de certo número de dentes. Adicionalmente, isolamento térmica, proteção contra impactos e vibrações e estanqueidade devem ser previstos no projeto do sistema de medição a fim de garantir sua robustez a qualquer meio de trabalho.

Além do volante do motor e do eixo piloto da transmissão, outros pontos do veículo podem ser usados para a medição da velocidade de rotação, como outras engrenagens da caixa de transmissão, dos eixos dianteiro e traseiro, do eixo cardã e do diferencial. O sistema desenvolvido aqui também pode ser utilizado em testes automotivos que requerem a medição da velocidade rotacional do motor com uma discretização melhor que a oferecida pelo barramento CAN. Outra aplicação interessante seria para o monitoramento de veículos de frota. Sensores poderiam ser instalados tanto em veículos de passeio como em veículos de carga para avaliar o modo de condução dos motoristas, ou seja, para detectar se eles trocam de marcha no momento certo. De forma mais ampla, este trabalho pode ser aplicado em qualquer sistema rotativo que contenha uma engrenagem, desde que respeitadas as condições para a instalação do sensor magnético.

REFERÊNCIAS

ADAMSON, S. Improved Approaches to the Measurement and Analysis of Torsional Vibration. **SAE World Congress**, Detroit, p. 23-31. jan. 2004.

ARM INFORMATION CENTER. **Cortex-A series processor**. Disponível em: <<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0344f/Beiibjca.html>>. Acesso em: 14 nov. 2015.

COUDERC, P. et al. Vehicle driveline behaviour: experimentation and simulation. **Journal Of Sound And Vibration**, Lyon, v. 218, n. 1, p. 133-157, 1998.

DION, J. et al. Gear impacts and idle gear noise: Experimental study and non-linear dynamic model. **Mechanical Systems And Signal Processing**, Saint Ouen, v. 23, n. 8, p. 2608-2628, 2009.

ELNADY, M. ; SINHA, J. ; OYADIJI, S. Identification of critical speeds of rotating machines using on-shaft wireless vibration measurement. **Journal Of Physics: Conference Series**, Manchester, v. 364, p. 1-10, 2012.

FLEMING, W. Overview of Automotive Sensors. **IEEE Sensors Journal**, Washington, v. 1, n. 4, p. 296-308, 2001.

IEEE COMPUTER SOCIETY (New York). **IEEE Standard for Local and metropolitan area networks: Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)**. 2011. Disponível em: <<https://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>>. Acesso em: 04 jun. 2015.

LECHNER, G. ; NAUNHEIMER, H. ; RYBORZ, J. **Automotives transmissions: fundamentals, selection, design, and application**. Berlim: Springer Verlag, 1999.

LISHA. **Welcome to the EPOS Project**. Disponível em: <<http://epos.lisha.ufsc.br/HomePage>>. Acesso em: 04 jun. 2015.

MIEYEVILLE, F. et al. Wireless sensor networks for active vibration control in automobile structures. **Smart Materials And Structures**, Lyon, v. 21, n. 7, p. 1-13, 2012.

QIN, G. ; HU, N. Design of Embedded Wireless Sensor and its Soft Encapsulation for Embedded Monitoring of Helicopter Planetary Gear Set. **Journal Of Physics: Conference Series**, Changsha, v. 364, p. 1-7, 2012

SARKAR, N. ; ELLIS, R. ; MOORE, T. Backlash detection in geared mechanisms: modeling, simulation, and experimentation. **Mechanical Systems And Signal Processing**, Honolulu, v. 3, n. 11, p. 391-408, 1997.

SINGH, R. ; XIE, H. ; COMPARIN, R. Analysis of automotive neutral gear rattle. **Journal Of Sound And Vibration**, Columbus, v. 131, n. 2, p. 177-196. 1989.

TURBTRON INDÚSTRIA E COMÉRCIO DE EQUIPAMENTOS ELÉTRICOS LTDA (Ribeirão Preto). **Pick-ups, Sensores Magnéticos Geradores de Impulsos**. 2001. Disponível em: <<http://www.turbtron.com.br/man/pickup.pdf>>. Acesso em: 04 jun. 2015.

WANG, P. et al. A torsional vibration measurement system. **IEEE Transactions On Instrumentation And Measurement**, West Lafayette, v. 41, n. 6, p. 803-807, 1992.

WHELAN, M. et al. Real-time wireless vibration monitoring for operational modal analysis of an integral abutment highway bridge. **Engineering Structures**, Potsdam, v. 31, p. 2224-2235, 2009.

APÊNDICE A

```

// Programa para recebimento de mensagens - master

#include <machine.h>
#include <alarm.h>
#include <mach/mc13224v/emote2_startup.h>
#include <system/config.h>
#include <utility/ostream.h>
#include <utility/string.h>
#include <gpio_pin.h>

__USING_SYS

struct Message
{
    unsigned short id_send;
    volatile TSC::Time_Stamp tx_value;
} Msg;

NIC * nic;
OStream cout;

void master()
{
    NIC::Protocol prot;
    NIC::Address src;

    while(true)
    {
        while(!(nic->receive(&src, &prot, &(Msg), sizeof(Msg)) > 0));
        if(Msg.tx_value < 1000)
        {
            cout << Msg.id_send << "0" << Msg.tx_value << endl;
        }
        else cout << Msg.id_send << Msg.tx_value << endl;
    }
}

int main()
{
    nic = new NIC();
    master();
}

```

APÊNDICE B

```

// Programa para leitura do sensor magnetico e envio de mensagens - slave
// Placa 1 - 1500 a 15500 Hz

#include <system/config.h>
#include <mach/mc13224v/emote2_startup.h>
#include <alarm.h>
#include <utility/ostream.h>
#include <chronometer.h>
#include <tsc.h>
#include <machine.h>
#include <utility/string.h>
#include <gpio_pin.h>

__USING_SYS

struct Message
{
    unsigned short id_send;
    volatile TSC::Time_Stamp tx_value;
} Msg;

NIC * nic;
Chronometer chrono;
OStream cout;

volatile TSC::Time_Stamp sensor_read = 0;
volatile int count = 0;

void timer_func(void);
void send_msg(unsigned short, volatile TSC::Time_Stamp);

int main()
{
    nic = new NIC();

    //timer 0 pin 8 - configura GPIO do timer 0 para funcionar como funcao alternativa:
    GPIO_Pin gpio(8, MC13224V_GPIO_Pin::FUNC_ALT1);
    //configura timer0 para ler da entrada (modo externo):
    MC13224V_Timer_0 timer(timer_func, MC13224V_Timer::EXT);

    //Inicializacao na faixa 1:
    int range = 1;
    int N_SAMPLES = 2;
    timer.comp1(0x5C); //92 em decimal
    chrono.start();

    while(1)

```

```

{
    if(count==N_SAMPLES)
    {
        switch(range)
        {
            case 1: //Faixa 1 - 1500 a 5000 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(11, sensor_read);
                if(sensor_read < 369) //Critério de mudança pra faixa 2
                {
                    timer.comp1(0x5E); //94 em decimal
                    range = 2; //Muda pra faixa 2
                }
                break;
            case 2: //Faixa 2 - 5000 a 6500 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(12, sensor_read);
                if(sensor_read > 376) //Critério de mudança pra faixa 1
                {
                    timer.comp1(0x5C); //92 em decimal
                    range = 1; //Muda pra faixa 1
                }
                if(sensor_read < 290) //Critério de mudança pra faixa 3
                {
                    timer.comp1(0x60); //96 em decimal
                    range = 3; //Muda pra faixa 3
                }
                break;
            case 3: //Faixa 3 - 6500 a 8000 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(13, sensor_read);
                if(sensor_read > 295) //Critério de mudança pra faixa 2
                {
                    timer.comp1(0x5E); //94 em decimal
                    range = 2; //Muda pra faixa 2
                }
                if(sensor_read < 241) //Critério de mudança pra faixa 4
                {
                    timer.comp1(0x66); //102 em decimal
                    range = 4; //Muda pra faixa 4
                }
            }
        }
    }
}

```

```

        break;
    case 4: //Faixa 4 - 8000 a 11000 Hz
        count = 0;
        sensor_read = chrono.ticks();
        chrono.reset();
        chrono.start();
        send_msg(14, sensor_read);
        if(sensor_read > 255) //Critério de mudança pra faixa 3
        {
            timer.comp1(0x60); //96 em decimal
            range = 3; //Muda pra faixa 3
        }
        if(sensor_read < 186) //Critério de mudança pra faixa 5
        {
            timer.comp1(0x68); //104 em decimal
            N_SAMPLES = 6;
            range = 5; //Muda pra faixa 5
        }
        break;
    case 5: //Faixa 5 - 11000 a 15500 Hz
        count = 0;
        sensor_read = chrono.ticks();
        chrono.reset();
        chrono.start();
        send_msg(15, sensor_read);
        if(sensor_read > 566) //Critério de mudança pra faixa 4
        {
            timer.comp1(0x66); //102 em decimal
            N_SAMPLES = 2;
            range = 4; //Muda pra faixa 4
        }
        break;
    }
}

}
}

void timer_func(void)
{
    count ++;
}

void send_msg(unsigned short id, volatile TSC::Time_Stamp value)
{
    Msg.id_send = id;
    Msg.tx_value = value;
    int r;
    while ((r = nic->send(NIC::BROADCAST, (NIC::Protocol) 1, &(Msg), sizeof(Msg)))
    != 11);
}

```

APÊNDICE C

```

// Programa para leitura do sensor magnetico e envio de mensagens - slave
// Placa 2 - 400 a 4133 Hz

#include <system/config.h>
#include <mach/mc13224v/emote2_startup.h>
#include <alarm.h>
#include <utility/ostream.h>
#include <chronometer.h>
#include <tsc.h>
#include <machine.h>
#include <utility/string.h>
#include <gpio_pin.h>

__USING_SYS

struct Message
{
    unsigned short id_send;
    volatile TSC::Time_Stamp tx_value;
} Msg;

NIC * nic;
Chronometer chrono;
OStream cout;

volatile TSC::Time_Stamp sensor_read = 0;
volatile int count = 0;

void timer_func(void);
void send_msg(unsigned short, volatile TSC::Time_Stamp);

int main()
{
    nic = new NIC();

    //timer 0 pin 8 - configura GPIO do timer 0 para funcionar como funcao alternativa:

    GPIO_Pin gpio(8, MC13224V_GPIO_Pin::FUNC_ALT1);
    //configura timer0 para ler da entrada (modo externo):
    MC13224V_Timer_0 timer(timer_func, MC13224V_Timer::EXT);

    //Inicializacao na faixa 1:
    int range = 1;
    int N_SAMPLES = 1;
    timer.comp1(0x48); //72 em decimal
    chrono.start();

```



```

while(1)
{
    if(count==N_SAMPLES)
    {
        switch(range)
        {
            case 1: //Faixa 1 - 400 a 1000 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(21, sensor_read);
                if(sensor_read < 722) //Critério de mudança pra faixa 2
                {
                    timer.comp1(0x52); //82 em decimal
                    range = 2; //Muda pra faixa 2
                }
                break;
            case 2: //Faixa 2 - 1000 a 2500 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(22, sensor_read);
                if(sensor_read > 820) //Critério de mudança pra faixa 1
                {
                    timer.comp1(0x48); //72 em decimal
                    range = 1; //Muda pra faixa 1
                }
                if(sensor_read < 329) //Critério de mudança pra faixa 3
                {
                    timer.comp1(0x56); //86 em decimal
                    N_SAMPLES = 2;
                    range = 3; //Muda pra faixa 3
                }
                break;
            case 3: //Faixa 3 - 2500 a 4133 Hz
                count = 0;
                sensor_read = chrono.ticks();
                chrono.reset();
                chrono.start();
                send_msg(23, sensor_read);
                if(sensor_read > 688) //Critério de mudança pra faixa 2
                {
                    timer.comp1(0x52); //22 em decimal
                    N_SAMPLES = 1;
                    range = 2; //Muda pra faixa 2
                }
                break;
        }
    }
}

```

```
        }  
    }  
}  
  
void timer_func(void)  
{  
    count ++;  
}  
  
void send_msg(unsigned short id, volatile TSC::Time_Stamp value)  
{  
    Msg.id_send = id;  
    Msg.tx_value = value;  
    int r;  
    while ((r = nic->send(NIC::BROADCAST, (NIC::Protocol) 1, &(Msg), sizeof(Msg)) !=  
11);  
}
```