

O SCILAB COMO FERRAMENTA DE AQUISIÇÃO DE DADOS E SIMULAÇÃO DE CONTROLE PID

SCILAB AS A TOOL FOR DATA ACQUISITION AND PID CONTROL SIMULATION

SCILAB COMO HERRAMIENTA PARA LA ADQUISICIÓN DE DATOS Y SIMULACIÓN DE CONTROL PID



<https://doi.org/10.56238/ERR01v10n4-017>

Victor Inacio de Oliveira

Doutor em Engenharia de Controle e Automação

Instituição: Escola Politécnica da USP, Universidade Presbiteriana Mackenzie, Faculdade Engenheiro Salvador Arena

Rogério Issamu Yamamoto

Doutor em Engenharia de Controle e Automação

Instituição: Escola Politécnica da USP, Universidade Presbiteriana Mackenzie, Faculdade Engenheiro Salvador Arena

Bruno Luis Soares de Lima

Doutor em Engenharia Elétrica

Instituição: Escola Politécnica da USP, Universidade Presbiteriana Mackenzie

Ricardo Janes

Doutor em Engenharia Elétrica

Instituição: Escola Politécnica da USP, Universidade Presbiteriana Mackenzie

Fabio Jesus Moreira de Almeida

Doutor em Engenharia de Materiais e Nanotecnologia

Instituição: Universidade Presbiteriana Mackenzie

Alex Lopes de Oliveira

Mestre em Engenharia Elétrica

Instituição: Escola Politécnica da USP, Universidade Presbiteriana Mackenzie

RESUMO

O artigo discute a relevância do controle PID (Proporcional, Integral e Derivativo) em processos industriais que requerem elevada precisão, rápida resposta e oscilações mínimas. O controle PID integra três estratégias de atuação, visando assegurar a estabilidade e o desempenho adequado do sistema, sendo amplamente empregado em contextos nos quais a tolerância a variações é reduzida. Ressalta-se que a etapa de aquisição de dados é fundamental para o êxito do controle, uma vez que falhas na coleta, no transporte ou na análise dos dados podem resultar em inconsistências, perdas e aumento de custos operacionais. Como exemplo prático, apresenta-se a comunicação entre a placa

Arduino AT MEGA 2560 e o software Scilab para a geração, leitura, tratamento e visualização de sinais, evidenciando a importância de uma base de dados sólida no desenvolvimento do controle PID. O sistema PID é detalhado em suas três componentes a proporcional, integral e derivative, que, de forma conjunta, promovem resposta rápida, minimização das oscilações e redução do erro, possibilitando a obtenção de precisão próxima ao valor de referência (set point) estabelecido. O estudo enfatiza que a implementação eficaz de um controle PID depende do emprego adequado dos blocos funcionais que compõem o controlador, de modo a compensar as limitações inerentes a cada componente individualmente. O objetivo central deste trabalho é demonstrar todas as etapas que antecedem a implementação do PID, destacando sua aplicabilidade em ambientes industriais que demandam sistemas de controle robustos e confiáveis.

Palavras-chave: Controle PID. Aquisição de Dados. Processos Industriais.

ABSTRACT

This article discusses the relevance of PID (Proportional, Integral, and Derivative) control in industrial processes that require high precision, fast response, and minimal fluctuations. PID control integrates three operating strategies to ensure system stability and adequate performance, and is widely used in contexts where tolerance to variations is limited. It is important to emphasize that the data acquisition stage is crucial to the success of the control, since failures in data collection, transportation, or analysis can result in inconsistencies, losses, and increased operating costs. As a practical example, the communication between the Arduino AT MEGA 2560 board and Scilab software for signal generation, reading, processing, and visualization is presented, highlighting the importance of a solid database in the development of PID control. The PID system is detailed in its three components: proportional, integral, and derivative, which, together, promote fast response, minimize fluctuations, and reduce error, enabling accuracy close to the established reference value (set point). The study emphasizes that effective implementation of PID control depends on the appropriate use of the functional blocks that make up the controller, compensating for the limitations inherent in each component individually. The main objective of this work is to demonstrate all the steps that precede PID implementation, highlighting its applicability in industrial environments that require robust and reliable control systems.

Keywords: PID Control. Data Acquisition. Industrial Processes.

RESUMEN

Este artículo analiza la relevancia del control PID (Proporcional, Integral y Derivativo) en procesos industriales que requieren alta precisión, respuesta rápida y mínimas fluctuaciones. El control PID integra tres estrategias operativas para garantizar la estabilidad del sistema y un rendimiento adecuado, y se utiliza ampliamente en contextos con tolerancia limitada a las variaciones. Es importante destacar que la etapa de adquisición de datos es crucial para el éxito del control, ya que fallos en la recopilación, el transporte o el análisis de datos pueden generar inconsistencias, pérdidas y un aumento de los costes operativos. Como ejemplo práctico, se presenta la comunicación entre la placa Arduino AT MEGA 2560 y el software Scilab para la generación, lectura, procesamiento y visualización de señales, destacando la importancia de una base de datos sólida en el desarrollo del control PID. El sistema PID se detalla en sus tres componentes: proporcional, integral y derivativo, que, en conjunto, promueven una respuesta rápida, minimizan las fluctuaciones y reducen el error, permitiendo una precisión cercana al valor de referencia establecido (punto de ajuste). El estudio enfatiza que la implementación efectiva del control PID depende del uso adecuado de los bloques funcionales que componen el controlador, compensando las limitaciones inherentes a cada componente individualmente. El objetivo principal de este trabajo es demostrar todos los pasos que preceden a la implementación del control PID, destacando su aplicabilidad en entornos industriales que requieren sistemas de control robustos y confiables.

Palabras clave: Control PID. Adquisición de Datos. Procesos Industriales.

100
110
100
10101 100
11 110
100 100
10 0101

100 100
110 110
100 100
10101 10101
11 10011
100 110100
101100 101
10101

1 INTRODUÇÃO

Muitos processos industriais e operacionais exigem controle preciso, resposta rápida e mínima oscilação devido à delicadeza dos sistemas ou à baixa tolerância à variação de sinal (Pandy et al, 2025). O controle PID (Proporcional, Integral e Derivativo) (Ogata, 2011) surge como solução eficiente para essas demandas, integrando três estratégias de controle que, quando combinadas, proporcionam estabilidade, precisão e desempenho superior, sendo indispensável em diversas aplicações. Atualmente com a inserção da inteligência artificial a precisão dos sistemas está se tornando cada vez mais necessária e técnicas de controle estão sendo cada vez mais exigidas (Pandy, Pugazhenthi, Murugan, 2025). Além disso, sistemas de aquisição de dados se tornam extremamente necessários no conceito de indústria 4.0, como por exemplo na implementação de gêmeos digitais (Duarte et al, 2025).

A elaboração de um sistema de controle PID eficaz depende, primeiramente, da correta aquisição de dados, etapa que envolve a coleta e o transporte das informações geradas, predominantemente, na planta de processo ou em ambientes controlados para avaliação de desempenho (Shah Ankit, Anilkumar, Patel, 2025). Este trabalho aborda tanto a origem dos dados quanto as técnicas de aquisição e transmissão, exemplificando diferentes cenários práticos.

Além disso, o trabalho detalha o desenvolvimento do sistema de controle PID, analisando separadamente os conceitos de controle proporcional, proporcional-integral e proporcional-derivativo (Felicio, 2010), bem como sua integração para obtenção de um controle robusto, de baixa oscilação e resposta ágil.

O objetivo central deste trabalho é demonstrar a comunicação entre a placa Arduino AT MEGA 2560 e o software Scilab (Amaral et al, 2015), visando a geração, leitura, tratamento e plotagem de sinais digitais e analógicos. Dessa forma, busca-se simular todas as etapas que antecedem a implementação de um sistema de controle PID, explorando suas funcionalidades e vantagens em processos que exigem alto desempenho e confiabilidade.

2 DESENVOLVIMENTO

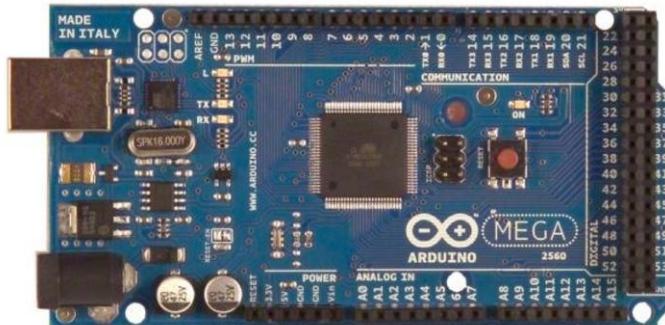
2.1 ARDUINO

O Arduíno MEGA2560 é uma plataforma didática de programação que utiliza linguagem C++. O modelo contempla alta utilidade em processos, já que possui 54 pinos (entradas e saídas) podendo ser utilizados 14 com PWM, 4 UARTS, permitindo energização por USB e fonte.

1 0 0
1 1 0
1 0 0
1 0 1 0 1 1 0 0
1 1 1 1 0 0
1 0 0 1 0 0
1 0 1 0 1

1 0 0 1 0 0
1 1 0 1 1 0
1 0 0 1 0 0
1 0 1 0 1 1 0 0
1 1 1 0 0 1 1 1
1 0 0 1 1 0 1 0 0
1 0 1 1 0 1 0 1
1 0 1 0 1

Figura 1 – Arduino MEGA 2560



Fonte: Arduino Datasheet (2015)

As grandes vantagens da plataforma são a facilidade e flexibilidade de aplicação. A programação é feita em seu microprocessador central, possuindo 256kB de memória flash para armazenar códigos. Seus pinos de saídas e entradas padronizam um valor de 5 V de utilização e um pino especial de 3,3 V, junto ao GND. Toda sua programação é feita através do próprio software Arduíno, que é disponível gratuitamente no site da Arduíno. As principais características dos componentes presentes na plataforma se apresentam na figura 2 abaixo e foram retiradas do datasheet do Arduino (Arduino datasheet, 2015).

Figura 2 – Características do Arduino ATmega2560.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Fonte: Arduino Datasheet (2015)

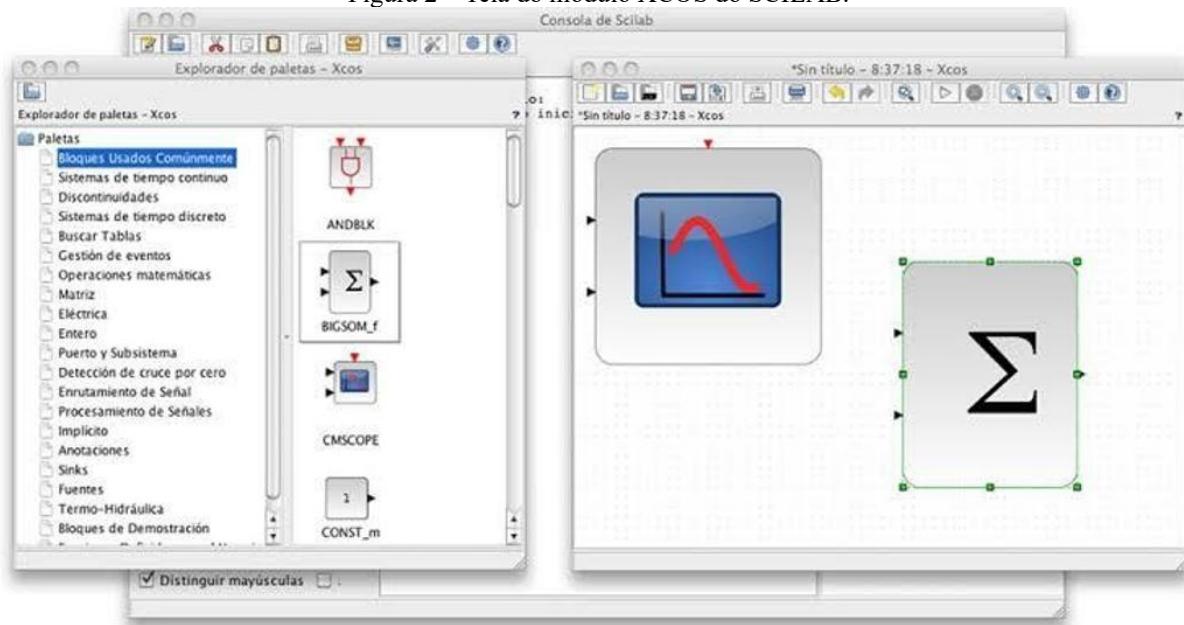
2.2 SCILAB

Scilab é um software de plataforma aberta para modelagem e prototipagem de sistemas de controle que tem ampla aplicação, principalmente em desenvolvimentos conjuntos com outros softwares como Arduíno, Python, entre outros (Filho, 2007).

O software foi desenvolvido em 1989 por um grupo de pesquisadores da INRIA (Institut National de Recherche en Informatique et en Automatique - Instituto Nacional de Pesquisa em Ciência da Computação e Automação) e da ENPC (École Nationale des Ponts et Chaussées – Escola Nacional de Pontes e Estradas). Admite como sistemas operacionais Linux, Windows, Solaris, Unix. Sua

configuração admite rotinas de linguagem FORTRAN ou C, o que permite sistemas de simulação e teste de controles, utilizando prévias de projetos que utilizam monitoramento sistemático. Além disso, o software disponibiliza plataforma de programação por blocos, e em códigos de linguagem, denominada XCOS. Na figura 3 a seguir é possível ver uma imagem de como é a tela de edição do XCOS.

Figura 2 – Tela do módulo XCOS do SCILAB.



Fonte: Malavida, 2010.

2.3 PROCEDIMENTO EXPERIMENTAL

O processo de aquisição de dados foi dividido em duas partes: leitura de sinal emitido da placa Arduino e visualização do mesmo em software Scilab e teste de controle proporcional (P), proporcional integral (PI), proporcional derivativo (PD) e proporcional integral derivativo (PID). Para realização dos exercícios propostos é necessário estabelecer uma comunicação entre o software Scilab e a placa Arduino AT Mega 2560.

O controle dos parâmetros é feito no próprio Scilab. Utilizando a ferramenta XCOS, que é editor de gráficos para modelagem sistemas dinâmicos em domínios de tempo discreto e contínuo. Nele os modelos podem ser projetados, carregados, salvos, compilados e simulados.

2.3.1 Etapa 1 - Estabelecer a comunicação entre o Arduino e SCILAB

A aquisição de dados para etapa 1 é designada para efetuar leitura de sinal emitido pela placa Arduino (nesta etapa é utilizado um arquivo que já possui os dados do sinal emitido pela placa Arduino

```

1 0 0
1 1 0
1 0 0
1 0 1 0 1 1 0
1 1 1 1 0
1 0 0 1 0 0
1 0 1 0 1 0 1

```

```

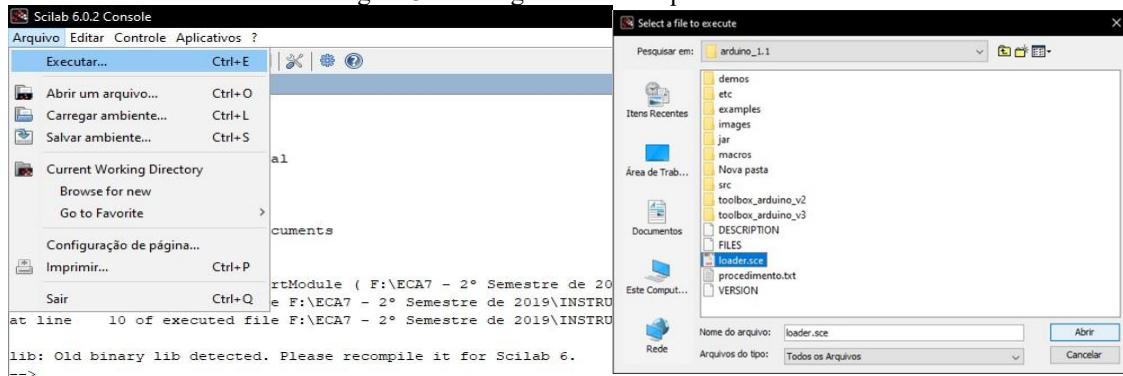
1 0 0 1 0 0
1 1 0 1 1 0
1 0 0 1 0 0
1 0 1 0 1 0 1
1 1 1 0 0 1 1
1 0 0 1 1 0 0 0
1 0 1 1 0 0 1 0
1 0 1 0 1

```

para gerar o sinal) e transmiti-lo para o software Scilab para que possa ser visualizado por meio de gráficos.

O primeiro passo para realizar a comunicação é executar o arquivo loader que possui os dados dos sinais que serão analisador e foram disponibilizado pelo docente dentro do Scilab, os passos podem ser vistos na figura 3 a seguir.

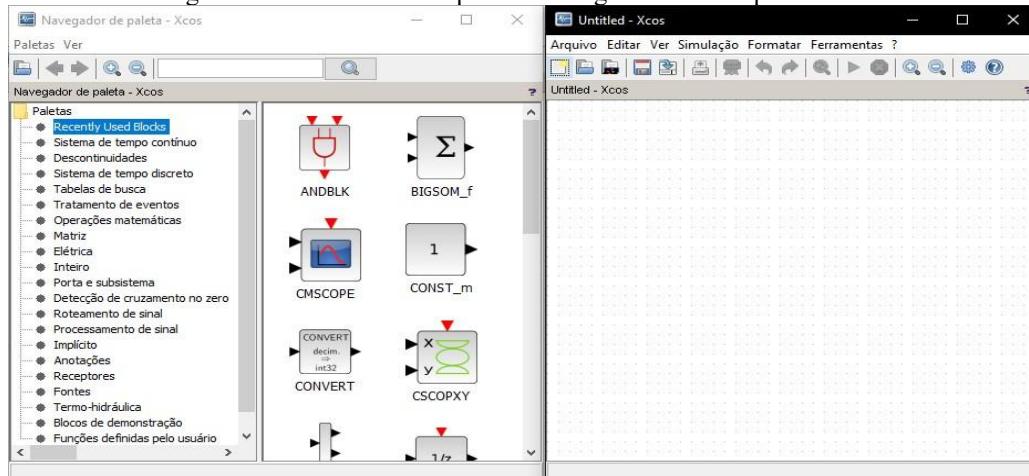
Figura 3 – Carregamento do arquivo loader.



Fonte: próprio autor, 2025.

Com o arquivo loader carregado corretamente, seguimos para a etapa de abrir o XCOS como demonstrado na figura 4.

Figura 4 – XCOS aberto depois do carregamento do arquivo loader.



Fonte: próprio autor, 2025.

Se a operação estiver correta, uma pasta com o nome de Arduino irá aparecer no canto esquerdo no final da pasta de paletas. A partir deste ponto é necessário abrir o software Arduino para carregar o programa “toolbox v3” que também foi disponibilizado pelo professor, esse programa será o responsável pelo envio do sinal e a conexão dos elementos escolhidos na montagem do XCOS com a programação.

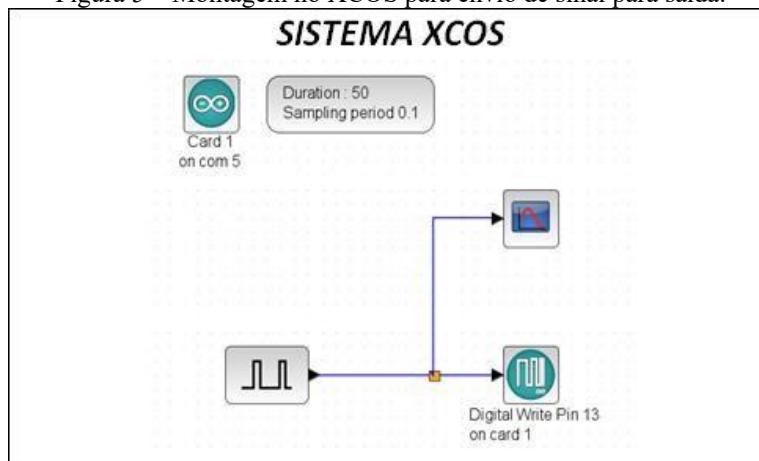
```
1 0 0      1 0 0
1 1 0      1 1 0
1 0 0      1 0 0
1 0 1 0 1  1 0 0
1 1 1 1 0  1 1 0 1 1
1 0 0 1 0  1 0 0 1 0
1 0 1 0 0  1 0 1 0 1
```

```
1 0 0      1 0 0
1 1 0      1 1 0
1 0 0      1 0 0
1 0 1 0 1  1 0 1 0 1
1 1 1 0 1  1 1 0 1 1
1 0 0 1 0  1 0 0 1 0 0
1 0 1 0 0  1 0 1 0 1
```

É importante observar em qual porta o Arduino se conectou antes de realizar o carregamento do programa devido ao limite da porta de acesso do Arduino, que para esta situação é no máximo a Porta COM9.

Depois do carregamento da programação do Arduino a próxima etapa é a montagem no XCOS dentro do Scilab. Os dispositivos utilizados são: Card1 e DigitalWrite, que podem ser encontrados na pasta componentes do Arduino; um CScope localizado na pasta de Fontes e Pulse_SC que se localiza na pasta de receptores. A figura 5 mostra o esquema montado no XCOS.

Figura 5 – Montagem no XCOS para envio de sinal para saída.

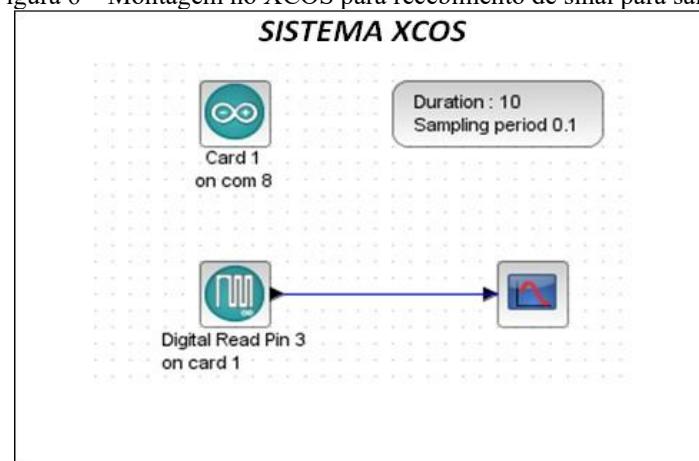


Fonte: próprio autor, 2025.

Para determinar os parâmetros basta realizar um duplo clique nos elementos do Arduino e do XCOS. O experimento em questão utilizou a Porta COM8 com duração de 10 e 20 segundos e período de amostragem de 0.1 e a leitura digital sendo feita no pino 13.

Logo após, com a comunicação estabelecida agora é feito o recebimento do sinal do Arduino para o Scilab utilizando a montagem como ilustra a figura 6.

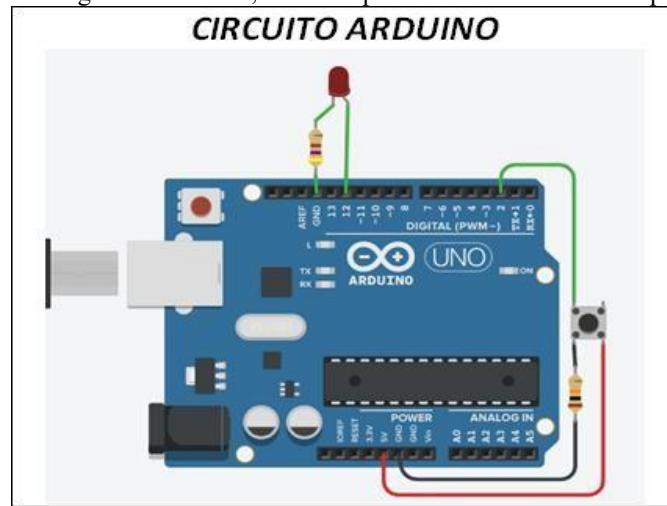
Figura 6 – Montagem no XCOS para recebimento de sinal para saída.



Fonte: próprio autor, 2025.

Na figura 7 é possível observar o circuito do Arduino montado para envio e recebimento de sinal. O envio é feito para o Led e o recebimento é feito pelo botão.

Figura 7 – Montagem do Arduino, utilizada para simular o envio e recepção de sinal.



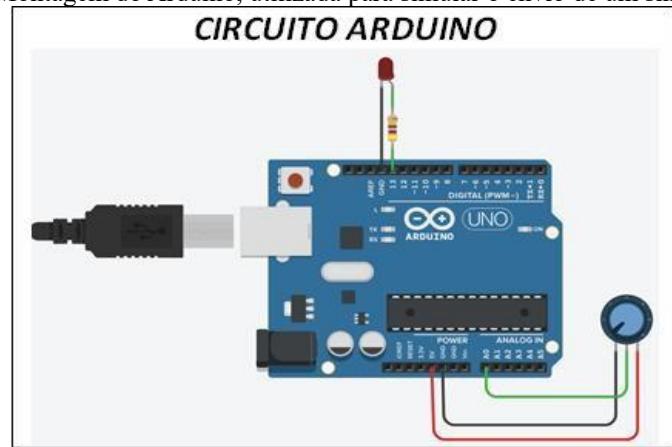
Fonte: próprio autor, 2025.

2.3.2 Teste de controle proporcional (P), proporcional integral (PI), proporcional derivativo (PD) e proporcional integral derivativo (PID)

O controle PID é aplicado para sistemas que necessitam de alta precisão com pouca variação de sinal e alta taxa de aproximação do valor de set point desejado. Para que essa meta seja estabelecida, o controle precisa ser dividido em proporcional, proporcional integral e proporcional derivativo. Com os três tipos de controle modelados pode-se dar andamento para o processo final, no qual todos são integrados gerando o sistema de controle PID.

Para o teste do controle PID foi utilizado uma entrada analógica do Arduino. Para se certificar que a entrada analógica estava respondendo corretamente foi utilizado um potenciômetro ligado ao Arduino AT Mega2560 o conectando a uma de suas entradas analógicas (A0), GND e 5 Volts para teste como demonstra a figura 8.

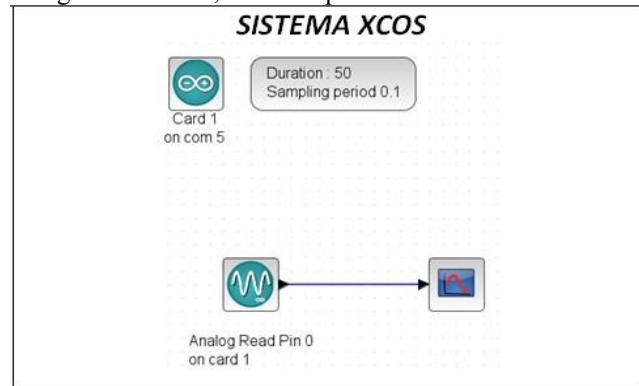
Figura 8 – Montagem do Arduino, utilizada para simular o envio de um sinal analógico.



Fonte: próprio autor, 2025.

Em resposta a isso também é necessário realizar um ajuste no XCOS para monitorar o comportamento do componente. Como ilustra a figura 9.

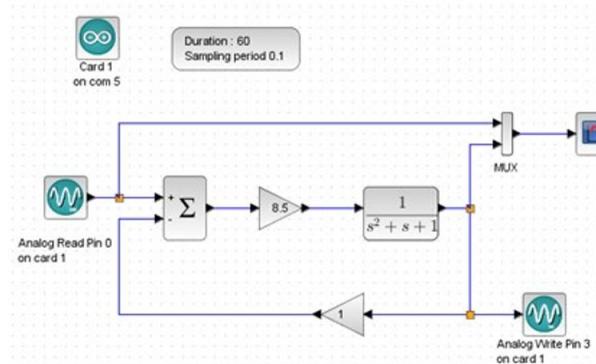
Figura 9 – Montagem no XCOS, utilizada para simular o envio de um sinal analógico.



Fonte: próprio autor, 2025.

A partir desse teste foi possível iniciar o procedimento de verificação dos controles P, PI, PD e PID. O controle proporcional integral (PI) tem como objetivo eliminar o erro que o controle P gera, diminuindo muito o erro entre o valor real e o set point. E para que essa aproximação seja feita é necessária aplicação de novos blocos de função como demonstra a figura 10. Novamente novas curvas são obtidas devido ao implemento dos blocos. Porém ainda há uma grande oscilação no sinal.

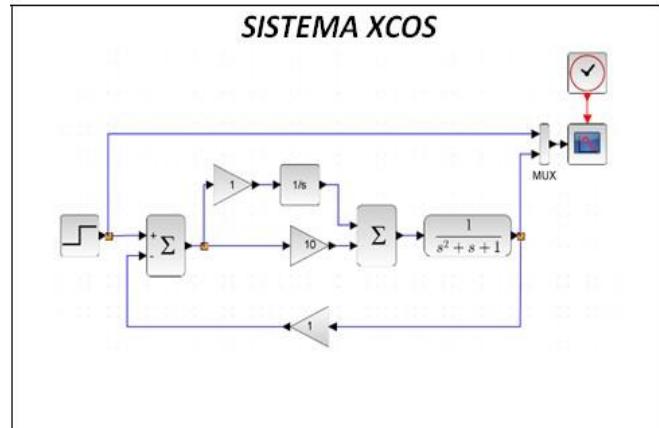
Figura 10 – Diagrama utilizado para simulação do bloco proporcional.
SISTEMA XCOS



Fonte: próprio autor, 2025.

Na sequência foi montado o diagrama para a simulação do PI. O controle proporcional integral (PI) tem como objetivo eliminar o erro que o controle P gera, diminuindo muito o erro entre o valor real e o set point. E para que essa aproximação seja feita é necessária aplicação de novos blocos de função como demonstra a figura 11. Novamente novas curvas são obtidas devido ao implemento dos blocos. Porém ainda há uma grande oscilação no sinal.

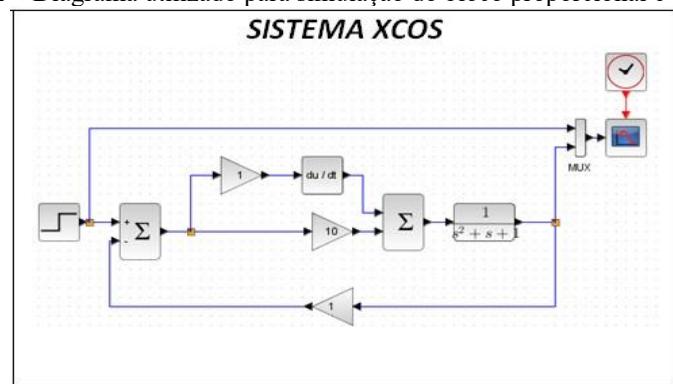
Figura 11 – Diagrama utilizado para simulação do bloco proporcional e integral.



Fonte: próprio autor, 2025.

Logo após foi feita a montagem para o diagrama PD (proporcional – derivativo). Para sistema que necessitam de pouca oscilação de sinal e de uma aproximação alta é utilizado o sistema de controle proporcional Derivativo (PD). A figura 12 mostra novos blocos a serem inseridos no sistema e com a manipulação do potenciômetro, observamos que perdemos a precisão de ajuste ao set point.

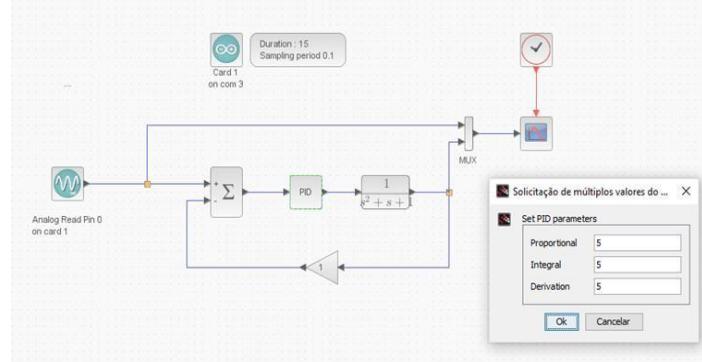
Figura 12 – Diagrama utilizado para simulação do bloco proporcional e derivativo.



Fonte: próprio autor, 2025.

Por fim, com todas as combinações já realizadas é o momento de integral os sistemas com o intuito de se obter um sistema de controle que possua as características positivas de cada um de modo que supram seus pontos negativos entre si. O controle proporcional integral derivativo (PID) possui como suas características baixa erro em comparação com o set point, alto taxa de aproximação e baixo sobressinal. Para que esse sistema seja obtido utilizaram-se os blocos de função demonstrados na figura 13.

Figura 13 – Diagrama utilizado para simulação do bloco proporcional, integral e derivativo.



Fonte: próprio autor, 2025.

É importante ressaltar que todas as etapas utilizaram a mesma montagem do Arduino da mostrada na figura 8 em diante e não ouve alteração pois, apenas com a manipulação do potenciômetro foram obtidas as curvas para análise.

3 RESULTADOS

Com o objetivo de realizar a análise prática de controle por instrumentação, obteve-se gráficos plotados automaticamente no Scilab com interação em tempo real, isso serviu como um meio de verificação da interação do controle, todas etapas foram realizadas o que permitiu saber os diversos

```

1 0 0
1 1 0
1 0 0
1 0 1 0 1 1 0
1 1 0
1 0 0 1 0 0
1 0 1 0 0 1 0 0
1 0 1 0 1

```

```

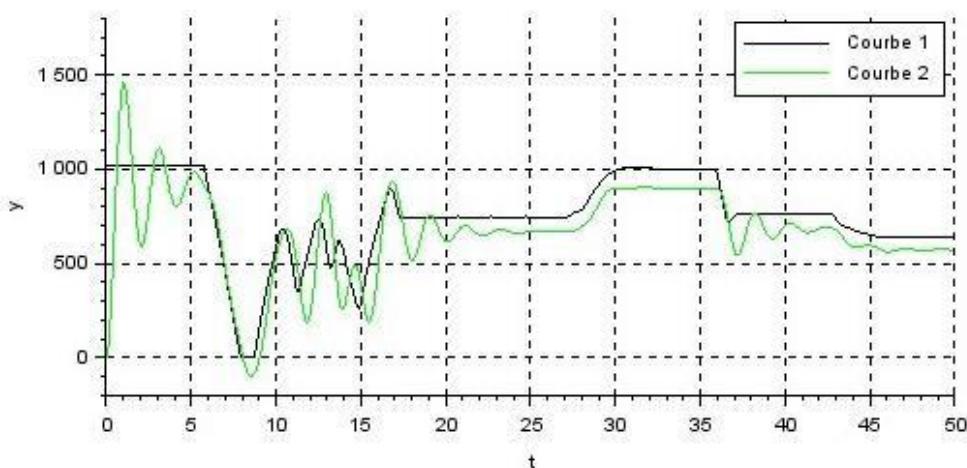
1 0 0 1 0 0
1 1 0 1 1 0
1 0 0 1 0 0
1 0 1 0 1 1 0 1
1 1 0 0 1 1 0 0
1 0 0 1 1 0 0 1 0
1 0 1 1 0 0 1 0 0
1 0 1 0 1

```

tipos de controle, com controle P, com controle PI e PD, PID, e sistemas com entrada linear sem controle.

Inicialmente foi analisado o controle Proporcional, o esperado foi uma determinada a oscilação com erro, mas com PV (variável de processo) fosse próximo de SP (set-point) em valor, como ajuste de SP utilizou-se o mesmo potenciômetro da etapa anterior, assim o Arduino tem a função de fazer esse controle proporcional de acordo com a entrada ajustada lida. Na figura 14 é possível ver o resultado, nota-se que a variável PV em verde, se aproxima de SP, mas ainda é possível ver o erro.

Figura 14 – Gráficos com resultado do controle proporcional.



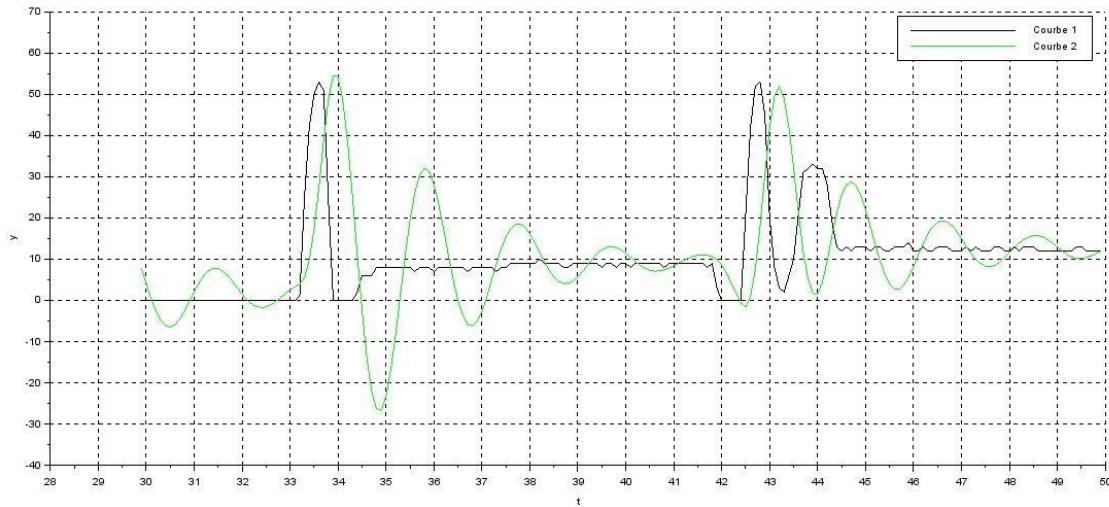
Fonte: próprio autor, 2025.

Com a análise do gráfico acima, percebe-se uma variação de PV sempre procura acompanhar SP devido o processamento do controlador, esse acompanhamento se dá devido o controle P (proporcional), além disso é possível verificar que o erro e a oscilação não foram eliminados. Todo esse processo foi dado por um K_p de 8.5.

Com a montagem vista na figura 12 se aplicou um sistema PD, ou seja, um controle Derivativo e Proporcional, adicionando o derivativo para diminuição da oscilação de resposta.

Assim, observou-se um sistema apenas com erro sem oscilação e que tenha boa proporcionalidade. Para isso adicionou-se um bloco amplificador para determinação de valor de K_d , e em série um bloco derivativo, em paralelo com isso utilizou-se o tradicional sistema de proporção usado na etapa anterior. A figura 15 mostra o resultado obtido nessa análise.

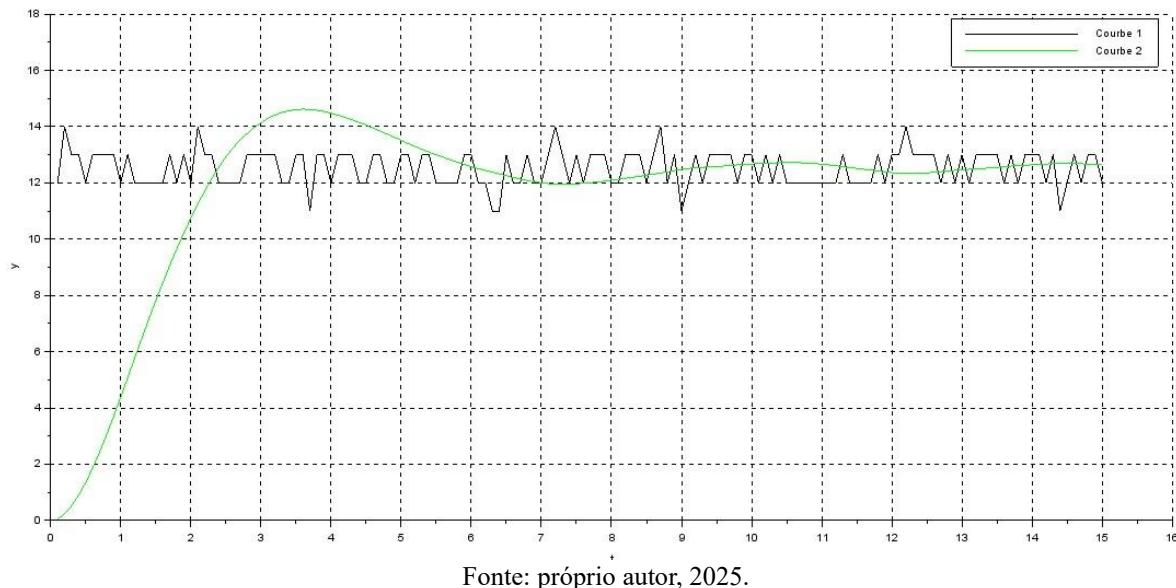
Figura 15 – Gráficos com resultado do controle proporcional e derivativo.



Fonte: próprio autor, 2025.

Com isso repare-se que a instantaneidade é mantida, o proporcional também, porém o derivativo elimina boa parte da oscilação quando comparado ao sistema puramente proporcional com o ajuste de kd é possível diminuir ainda mais a oscilação.

A última etapa propõe a junção dos tipos de controles, PID. Desta maneira ficou em aberto a montagem do diagrama de blocos, sugerindo como poderia ser feito o controle destes três coeficientes. No primeiro teste foi ajustado para $kp, kd, ki = 1$, a resposta pode ser vista na figura 16.

Figura 16 – Gráficos com resultado do PID com $kp, kd, ki = 1$.

Fonte: próprio autor, 2025.

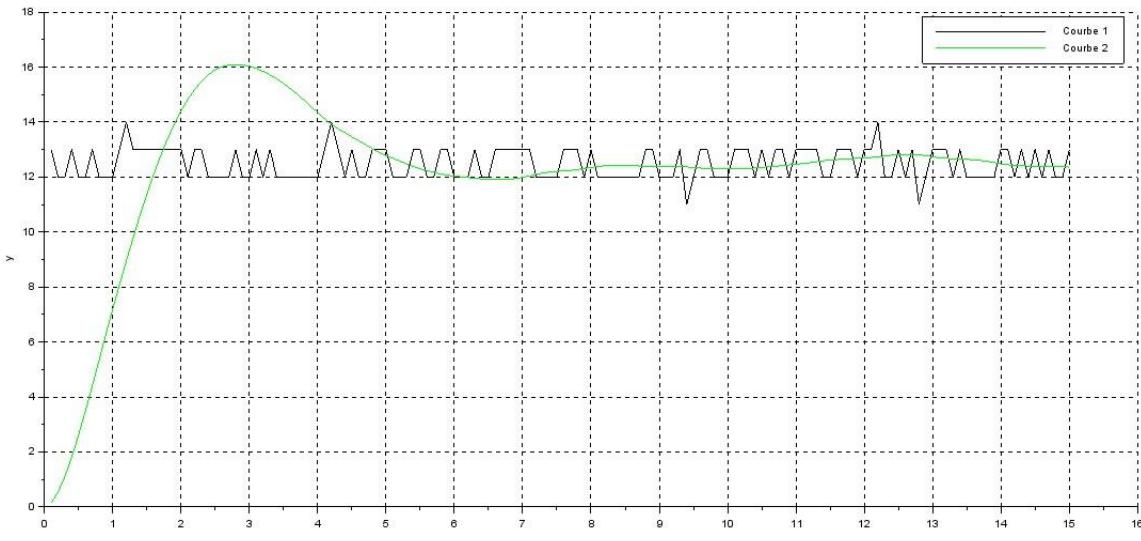
Verificou-se baixa oscilação e um erro bem menor, um controle preciso em que PV procura SP de forma rápida eliminando erros e oscilações dos processos. Logo após, foi feito o teste para $kp, kd, ki = 2,5$. O resultado pode ser visto na figura 17.

```

1 0 0
1 1 0
1 0 0
1 0 1 0 1 1 0 1 0 0
1 1 1 1 0 1 1 0 0
1 0 0 1 0 0
1 0 1 0 0 1 0 1
  
```

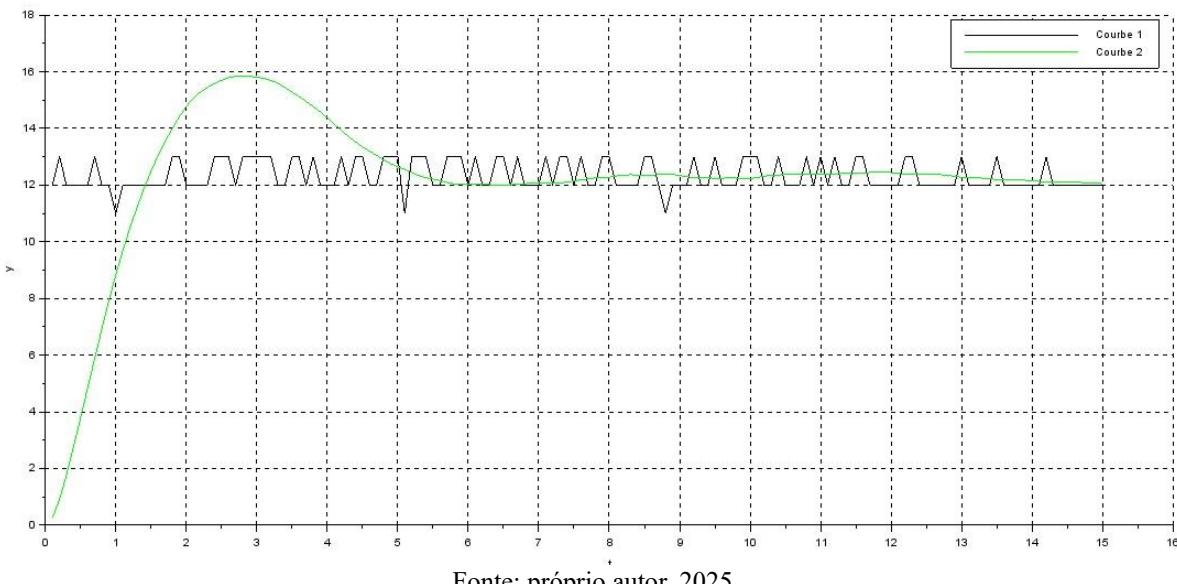
```

1 0 0 1 0 0
1 1 0 1 1 0
1 0 0 1 0 0
1 0 1 0 1 1 0 1 0 0
1 1 1 0 0 1 1 0 1 1
1 0 0 1 1 0 1 0 0
1 0 1 1 0 0 1 0 1
  
```

Figura 17 – Gráficos com resultado do PID com $k_p, k_d, k_i = 2,5$.

Fonte: próprio autor, 2025.

Verificou-se uma pequena melhoria na oscilação e erro. Agora foi feito o teste para $k_p, k_d, k_i = 5$, na figura 18 é possível ver o resultado.

Figura 17 – Gráficos com resultado do PID com $k_p, k_d, k_i = 5$.

Fonte: próprio autor, 2025.

Nota-se um erro quase nulo, e uma oscilação ideal já no sistema.

4 CONCLUSÕES

Conclui-se que a aquisição de dados é uma das fases mais importante para que o controle de processo possa ser feito de maneira correta, pois, se a coleta de dados for feita de maneira incorreta, assim como seu tratamento e análise acarretará em uma série de problemas, como inconstância de

```

1 0 0
1 1 0
1 0 0
1 0 1 0 1 1 0 0
1 1 1 1 0 0
1 0 0 1 0 0
1 0 1 0 0 1 0 0
1 0 1 0 1
  
```

```

1 0 0 1 0 0
1 1 0 1 1 0
1 0 0 1 0 0
1 0 1 0 1 1 0 0
1 1 1 0 0 1 1 0
1 0 0 1 1 0 1 0 0
1 0 1 1 0 0 1 0 0
1 0 1 0 1
  
```

dados, produtos não conformes, perdas de matéria prima e tempo e outros fatores que geram prejuízos e custos.

Deste modo é importante uma sólida e sucinta coleta, transporte, análise e apresentação de dados para que o controle de processo seja feito da maneira correta. Foi possível observar essa importância durante o desenvolvimento da comunicação entre Arduino e Scilab, que se feita de forma incorreta, impossibilita o desenvolvimento do PID.

O sistema PID como já mencionado anteriormente é consiste em três ilhas de controle: proporcional; proporcional integral e proporcional derivativo que juntas apresentam sistema de rápida resposta (P), baixa oscilação, o que evita grandes sobressinais ou até mesmo os extingue (PD) e a atenuação de erro que se torna no máximo de dois por cento para mais ou para menos de acordo com o set point definido.

E para que ele seja gerado é necessário muito blocos de funções que supram as desvantagens uns dos outros utilizando suas vantagens ou aplicabilidade.

REFERÊNCIAS

AMARAL, T.R., LEITE, N.M.G., LADEIA, V.F.A.C., CARVALHO, R.S., Modelagem e simulação de sistemas dinâmicos com o Silab/Xcos: um caderno didático, Anais do 6º Simpósio Hipertexto e Tecnologias na Educação, Pernambuco, 2015.

DUARTE IZIDORO, Julio Cesar; PARMIGIANI, Rebecca Vitoretti; GELTZENAUER, Felipe Taguetto; ANDREOLI, Camila Tvardoski; AUGUSTO, Lucas Demetrios; YAMAMOTO, Rogério Issamu; OLIVEIRA, Victor Inácio de. Desenvolvimento de um gêmeo digital com Raspberry Pi: uma abordagem acessível para integração entre ambientes físico e virtual. Revista Delos, [S.l.], 24 abr. 2025.

FELICIO, L.C.. Modelagem da dinâmica de sistemas e estudo da resposta. São Paulo: Rima, 2010.

FILHO, P.T.S., CARVALHO, M.O.M., VILLA, C.V.S., Utilização de ferramenta computacional para o ensino da dinâmica em engenharia mecânica, Anais do XXXV Congresso Brasileiro de Educação em Engenharia, Curitiba, 2007.

OGATA, K. Engenharia de controle moderno. 5a. ed., Pearson, São Paulo, 2011.

PANDY G., PUGAZHENTHI V. J., MURUGAN A., Generative AI: Trans-forming the Landscape of Creativity and Automation, Int. J. Comput. Appl., vol. 186, no. 63, pp. 07–13, Jan. 2025.

PANDY, G. et al, Instrumentation for Robotics and Automation: Innovations, Challenges, and Future Directions, 2025 11th International Conference on Automation, Robotics, and Applications (ICARA), Zagreb, Croatia, 2025, pp. 127-133, doi: 10.1109/ICARA64554.2025.10977712.

SHAH ANKIT K., M. ANILKUMAR M., PATEL H. R., Experimental Insights into the Modeling and Control of Heater Systems Using Discrete PID Controllers, 2025 International Conference on Sustainable Energy Technologies and Computational Intelligence (SETCOM), Gandhinagar, India, 2025, pp. 1-5, doi: 10.1109/SETCOM64758.2025.10932433.

