



Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Diseño y Desarrollo de un Prototipo de
Simulación para Robots Aéreos basado
en Unreal 5, ROS2 y Gazebo (Informe
Intermedio)**

Autor: Daniel Corrales Falco
Tutor(a): Santiago Tapia Fernandez

Madrid, Abril 2023

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática

Título: Diseño y Desarrollo de un Prototipo de Simulación para Robots Aéreos basado en Unreal 5, ROS2 y Gazebo

Madrid, Abril 2023

Autor: Daniel Corrales Falco

Tutor: Santiago Tapia Fernandez

Dpto. Lenguajes de Sistemas Informáticos e Ingeniería del Software
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Aquí va el resumen del TFG. Extensión máxima 2 páginas.

Agradecimientos

Gracias

Índice general

1. Introducción	1
1.1. Descripción General	1
1.2. Unreal Engine 5	2
1.3. ROS 2	2
2. Desarrollo	5
2.1. Descripción General	5
2.2. Fase de prototipado	5
2.2.1. Pruebas de movimiento de actores:	5
2.2.2. Prueba de importación de mallas:	12
2.2.3. Prueba de implementación de un control diferencial de la velocidad:	13
3. Title of next chapter	19
3.1. Overview	19
3.2. The next section	19
4. Title of next chapter	21
4.1. Overview	21
4.2. The next section	21
A. Title of the first appendix chapter	23
A.1. Overview	23
A.2. The next section	23
Bibliografía	25

1. Introducción

1.1. Descripción General

La simulación de drones es un campo relativamente nuevo, sin embargo, es un área en el que se han realizado numerosos desarrollos. En los recientes años se han realizado grandes avances e investigaciones con diversos fines, desde el estudio de la física del movimiento de los drones para su posterior simulación en entornos virtuales, hasta estudios para descubrir las posibles aplicaciones de estas versátiles máquinas. Como por ejemplo su uso en seguridad[2], mantenimiento de campos de placas solares[3] e incluso entretenimiento con enjambres de cientos de estos pequeños robots[1] .

Los drones son vehículos aéreos no tripulados que se crearon en un inicio con fines militares. Sin embargo, con el paso del tiempo se han encontrado una amplia variedad de usos como mencionaba anteriormente. Es esta diversificación y popularización de los drones la que nos lleva a querer utilizarlos en problemas de la vida cotidiana. Por ello, simular el comportamiento de una máquina de este tipo se ha vuelto esencial, tanto para prever que movimientos será capaz de realizar según en que condiciones se encuentre como para poder hacer pruebas sin el equipamiento real, evitando así posibles daños. Con esto en mente, se crean los simuladores, que son entornos virtuales tridimensionales los cuales pueden recrear una gran variedad de situaciones.

Con respecto a los avances actuales en el campo de los simuladores, se han desarrollado una gran cantidad de los mismos, entre ellos, los que permiten aprender a controlar un dron, otros orientados a videojuegos de simulación de carreras de drones[4], etc. A nivel más profesional podemos encontrar algunos como AirSim[5], simulador de vuelo creado por Microsoft en el motor de juego de Unreal Engine (UE), o Flightmare[6], un simulador desarrollado para el motor de juego de Unity. Sin embargo, este simulador se basa en el simulador de físicas Gazebo, el cual para ciertas ocasiones es muy limitado.

La mayoría de los simuladores previamente descritos hacen uso de 2 SDK (Software Development Kit) externos, Gazebo y ROS/ROS2. El primero es un motor de físicas el cual gestiona todo lo relacionado con la física que interactúa con el dron, ya sea su movimiento, velocidad...etc. El segundo, ROS, es un software que se centra en el intercambio de mensajes, más concretamente, se encarga de enviarle las instrucciones al controlador del dron para manejar al mismo. Le envía datos como por ejemplo el modo de vuelo, velocidad y demás. Estos modos de vuelo y datos pueden variar dependiendo de que tipo de dron y sobre que tipo de software este construido.

Así pues, este trabajo tiene como objetivo crear un simulador de vuelo de drones en el entorno gráfico de Unreal Engine 5 realizando una integración con la librería de comunicación de C, ROS2, para así crear un sistema de manejo automático de la trayectoria del dron. La implementación de la gran mayoría del proyecto se realizará en C++. Al mismo tiempo, al crear este simulador en UE5, lo que se quiere es evitar el uso de Gazebo como motor de físicas ya que este es un tanto limitado. En cambio, se pretende simular la física y colisiones del dron con el propio motor de físicas de UE5.

1.2. Unreal Engine 5

Como he mencionado previamente, el simulador se va a desarrollar para el motor de juego Unreal Engine 5 (UE5) desarrollado por la compañía Epic Games. Esta herramienta es muy reciente y cuenta con unos avances gráficos enormes, pudiendo llegar a generar entornos que lucen casi idénticos a los reales dando la impresión de ser grabaciones del entorno natural y no simulaciones generadas por ordenador. Asimismo cuenta con un sistema de físicas y colisiones integrado, lo que facilitará las tareas de implementación más adelante. Como añadido, este entorno también cuenta con herramientas de Inteligencia Artificial, que se pueden usar para controlar los actores, refiriendonos a los drones, que podamos llegar a tener en la escena.

El principal motivo para realizar el desarrollo en este motor, es su gran proyección de futuro, gracias a la capacidad gráfica que proporciona y la oportunidad de hacer uso de IA, hacer el entrenamiento de los drones dentro del simulador para luego poder transferirlo a máquinas reales y así no arriesgarse a dañar el equipo.

1.3. ROS 2

ROS 2, o también conocido como *Robot Operating System 2*, es un SDK (*System Development Kit*) open source, el cuál ofrece una plataforma estándar para desarrollar software de cualquier rama de la industria que implique el uso de robots. Este framework se desarrolló en 2007 por el Laboratorio de Inteligencia Artificial de Standford y su desarrollo se ha continuado desde entonces.

La versión de este framework con la que estamos trabajando es la versión Humble, la última versión publicada a la fecha de realización de este trabajo. Entrando a describir más específicamente en que consiste este software, ROS se compone de 2 partes básicas, el sistema operativo ros, y ros-pkg, un conjunto de paquetes creados por la comunidad que implementan diversas funcionalidades como puede ser: localización, mapeo simultáneo, planificación, percepción y simulación...etc. Sin embargo, el uso principal de este conjunto de librerías es el paso de mensajes entre un controlador y la máquina en cuestión.

En este trabajo, el objetivo, además de poder tener un dron cuyo movimiento sea lo más fiel a la realidad posible, es implementar ROS 2 para poder realizar

1.3 ROS 2

el control del dron de forma “externa”, y así simular un vuelo real.

2. Desarrollo

2.1. Descripción General

Como se mencionó en la introducción, el desarrollo de esta aplicación será principalmente en C++ dado que es el lenguaje en el que se escribe en Unreal Engine 5 y ROS2. La fase de desarrollo se basa en 2 partes principales. Una primera de puesta a punto, o de prototipado, donde se ha experimentado con el motor de juego para descubrir la mejor forma de implementar los requisitos pedidos. Y una segunda fase en la que, ya , en esta primera versión se desarrolla la aplicación al completo.

2.2. Fase de prototipado

Esta ha sido la primera fase del desarrollo del simulador, en la cual he realizado algunas pruebas para encontrar la mejor forma de implementar un dron y su movimiento en el *sandbox* proporcionado por Unreal. Las pruebas realizadas han sido un total de 3 en relación a los elementos que personalmente he considerado más importantes para crear un simulador: el movimiento, tener una malla, o *mesh*, con la forma del dron y finalmente el control de la velocidad que usaremos para controlar la posición del dron.

2.2.1. Pruebas de movimiento de actores:

Para comenzar, se analizaron las posibilidades de movimiento de actores dentro de Unreal. En este aspecto cabe destacar que Unreal ofrece una serie de clases propias que nos permiten crear elementos que situar en la escena a los cuales añadirles componentes. Estas clases son: actor, peón y personaje, donde cabe decir que las dos últimas heredan de actor, lo cual nos indica que todas las funciones de actor se encuentran disponibles tanto en peón y personaje. Estos elementos son los objetos que podemos añadir a un nivel o escena creado en unreal, es decir son los elementos que interactuarán con el entorno y el usuario una vez comienza el juego. A pesar de que todas sean clases casi hermanas, las principales diferencias entre ellas son en relación a qué utilidad de Unreal son capaces de acceder. Los actores son los objetos más limitados ya que no ofrecen opción a que los maneje la IA ni permiten tener inputs de teclado. Los peones son el siguiente paso en la cadena, los cuales permiten ser controlados por la IA de Unreal y permiten inputs de teclado, sin embargo no tienen físicas integradas. Finalmente, los personajes son actores creados principalmente para

simular personas, estos incluyen todo lo mencionado previamente y además, integran las físicas necesarias para simular un humano.

En cuanto a los componentes que se pueden añadir a estos actores, son elementos que añaden funcionalidad a los mismos. Podemos encontrar 2 tipos de componentes: componentes de actor y componenetes de escena. El componente de actor no hace nada esencialmente, sin embargo es imprescindible dado que es la base de la jerarquía del resto de componentes. Los componentes de escena son aquellos que podemos usar como apoyo para incluir funcionalidades a los actores.

Ahora, todos estos elementos se integran en el editor de Unreal mediante jerarquías donde predominan las acciones que tome el componente raíz de la misma.

(foto de las jerarquías del editor en Unreal)

Haciendo uso de estos componentes diseñamos un primer prototipo para estudiar el movimiento en los actores y planificar la futura implementación del movimiento de un dron. Tras investigar y revisar tutoriales de diversas fuentes[tutorial de youtube y página de unreal] decidí crear un componente de escena que se encargara de modificar la localización de una *mesh* (malla). Es decir, crear un componente de escena que fuera el raíz de la jerarquía que moviera un modelo 3. En este caso, un cubo cuyo movimiento se basa en un vector de posición al cual se le podían ajustar las componentes tanto desde el editor como desde un *blueprint* como desde el propio código fuente.

A continuación abordamos la cuestión de las colisiones, añadiendo la propiedad comunmente conocida como *Hitbox* (caja de colisiones). Para esto simplemente añadí un componente de escena de tipo hitbox, lo ajusté al tamaño de la mesh elegida y configuré un gráfico de eventos para que cuando el jugador entrara en contacto con el objeto este empezara a moverse. Asimismo incluí otra Hitbox para delimitar un área amplia alrededor de la malla para que en el caso de que detectara que el jugador abandonaba el área el objeto volviera a su posición original.

A continuación se adjuntan los ficheros “move.h” y “move.cpp” que contienen el código generado para este componente:

Listing 2.1: Header del componente de escena move

```

1 //Daniel Corrales 2023
2 //TFG UPM
3
4 #pragma once
5
6 #include "CoreMinimal.h"
7 #include "Components/SceneComponent.h"
8 #include "move.generated.h"
9
10 UDELEGATE(BlueprintAuthorityOnly)
11 DECLARE_DYNAMIC_MULTICAST_DELEGATE_OneParam(FOnMoveReachEndPointSignature, bool,
12     IsTopEndpoint);
13
14 UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
15 class MOVETEST_API Umove : public USceneComponent
16 {

```

2.2 Fase de prototipado

```
16     GENERATED_BODY()
17
18 public:
19     // Sets default values for this component's properties
20     Umove();
21
22     UFUNCTION(BlueprintCallable)
23     void EnableMovement(bool shouldMove);
24
25     UFUNCTION(BlueprintCallable)
26     void ResetMovement();
27
28     UFUNCTION(BlueprintCallable)
29     void SetMoveDirection(int Direction);
30
31
32 protected:
33     // Called when the game starts
34     virtual void BeginPlay() override;
35
36 public:
37     // Called every frame
38     virtual void TickComponent(float DeltaTime, ELevelTick TickType,
39                               FActorComponentTickFunction* ThisTickFunction) override;
40
41 private:
42
43     //Offset to move
44     UPROPERTY(EditAnywhere)
45     FVector MoveOffset;
46
47     //Speed
48     UPROPERTY(EditAnywhere)
49     float Speed = 1.0f;
50
51     //Enable the movement of the component
52     UPROPERTY(EditAnywhere)
53     bool MoveEnable = true;
54
55     //On extreme reached event
56     UPROPERTY(BlueprintAssignable)
57     FOnMoveReachEndPointSignature OnEndpointReached;
58
59     //Computed locations
60     FVector StartRelativeLocation;
61     FVector MoveOffsetNorm;
62     float MaxDistance = 0.0f;
63     float CurDistance = 0.0f;
64     int MoveDirection = 1;
65
66 };
```

Listing 2.2: Código fuente del componente de escena move

```
1 //Daniel Corrales 2023
2 //TFG UPM
3
4
5 #include "move.h"
6
7
8 // Sets default values for this component's properties
9 Umove::Umove()
10 {
11     // Set this component to be initialized when the game starts, and to be
```

```

        ticked every frame. You can turn these features
12     // off to improve performance if you don't need them.
13     PrimaryComponentTick.bCanEverTick = true;
14
15     // ...
16 }
17
18 void Umove::EnableMovement(bool shouldMove)
19 {
20     // Assing value and set correct tick enable state
21     MoveEnable = shouldMove;
22     SetComponentTickEnabled(MoveEnable);
23 }
24
25 void Umove::ResetMovement()
26 {
27     //Clear distance and set to origin
28     CurDistance = 0.0f;
29     SetRelativeLocation(StartRelativeLocation);
30 }
31
32 void Umove::SetMoveDirection(int Direction)
33 {
34     MoveDirection = Direction >= 1 ? 1 : -1;
35 }
36
37 // Called when the game starts
38 void Umove::BeginPlay()
39 {
40     Super::BeginPlay();
41
42     // Set start location
43     StartRelativeLocation = this->GetRelativeLocation();
44
45     //Compute normalized movement
46     MoveOffsetNorm = MoveOffset;
47     MoveOffsetNorm.Normalize();
48     MaxDistance = MoveOffset.Size();
49
50     //Check if ticking is required
51     SetComponentTickEnabled(MoveEnable);
52 }
53
54
55
56 // Called every frame
57 void Umove::TickComponent(float DeltaTime, ELevelTick TickType,
58     FActorComponentTickFunction* ThisTickFunction)
59 {
60     Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
61
62     //Set the current distance
63     if(MoveEnable)
64     {
65         CurDistance += DeltaTime * Speed * MoveDirection;
66         if(CurDistance >= MaxDistance || CurDistance <= 0.0f){
67             //Inver direction
68             MoveDirection *= -1;
69
70             //Fire event
71             OnEndpointReached.Broadcast(CurDistance >= MaxDistance);
72
73             //Clamp distance
74             CurDistance = FMath::Clamp(CurDistance, 0.0f, MaxDistance);
75         }
76     }

```


2.2 Fase de prototipado

```
77
78 // Compute and set current location
79 SetRelativeLocation(StartRelativeLocation + MoveOffsetNorm * CurDistance);
80 }
```

Adjunto también el gráfico de eventos creado para completar su funcionalidad:

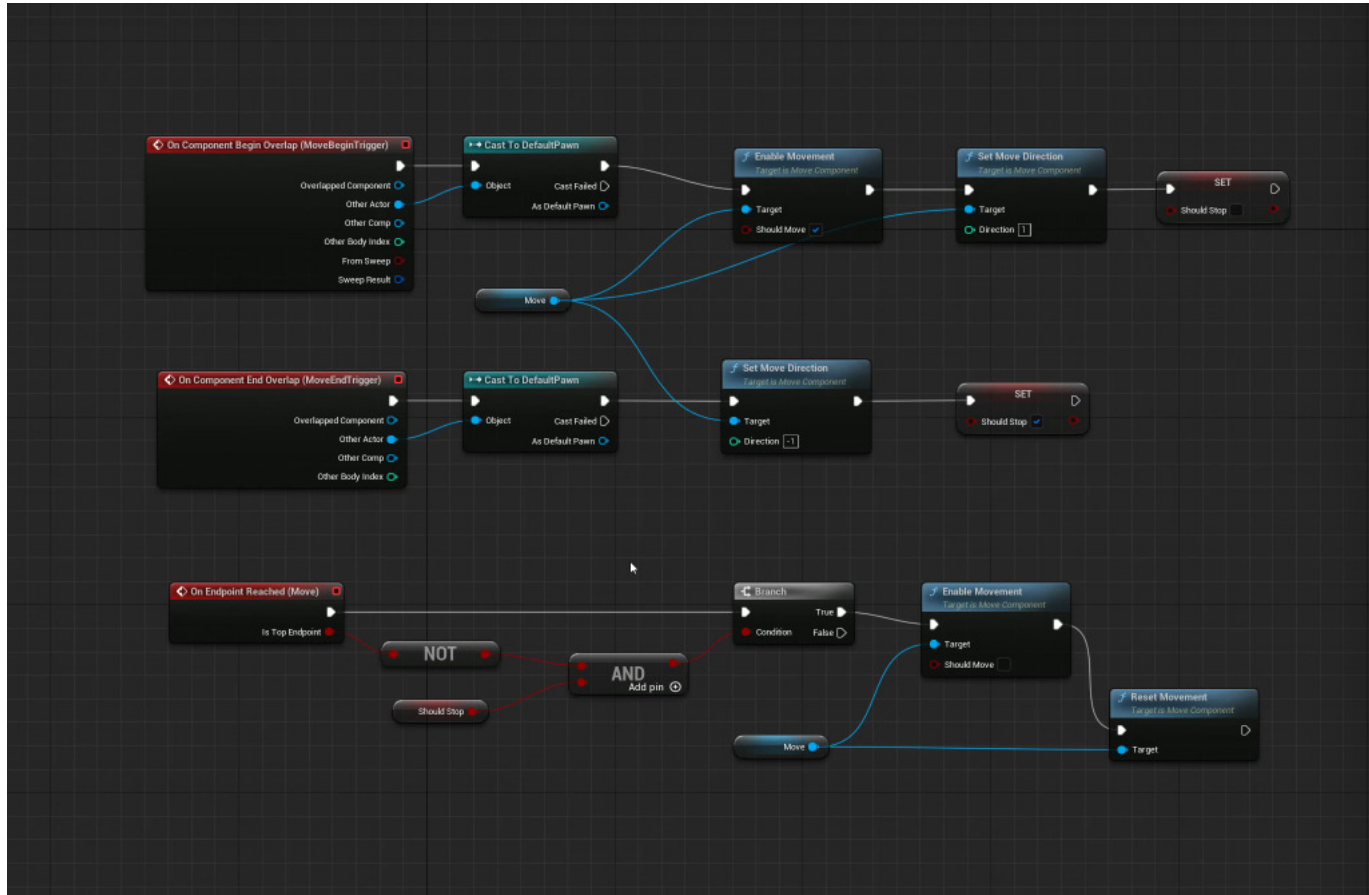


Figura 2.1.: Gráfico de eventos de activación de movimiento por colisión de hitbox

El código y el gráfico de eventos fue obtenido de la guía creada por Lötwig Fusel [referencia que toque]

Posteriormente, tras este primer acercamiento a los actores y componentes, decidí realizar pruebas para intentar implementar un modo de vuelo típico de los drones, el modo *Hover*. Este modo es un modo simple en el que el dron se eleva hasta una altura pre establecida y la mantiene hasta que se le envíe una señal para moverse o volver a su posición original.

Esta prueba es muy similar a la anterior dado que también se basa en un componente de escena, sin embargo, hay algún cambio respecto al funcionamiento. El código para modificar la posición sigue siendo el mismo, solo que en este caso, podemos modificar únicamente la altura máxima que queremos que alcance el dron y la velocidad del mismo. También se implementaron dos funciones, una

para comenzar el vuelo y otra para resetearlo y cabe mencionar que esta vez no se ha implementado ningún grafico de eventos dado que la interacción de esta prueba se basa en mandar la señal de activación del hover o de reset.

Adjunto tanto el header como el código del componente en cuestión “*HoverComponent*”:

Listing 2.3: Header del componente de escena HoverComponent

```

1 //Daniel Corrales 2023
2 //TFG UPM
3
4 #pragma once
5
6 #include "CoreMinimal.h"
7 #include "Components/SceneComponent.h"
8 #include "HoverComponent.generated.h"
9
10
11 UCLASS( ClassGroup=(Custom), meta=(BlueprintSpawnableComponent) )
12 class HOVERTEST_API UHoverComponent : public USceneComponent
13 {
14     GENERATED_BODY()
15
16 public:
17     // Sets default values for this component's properties
18     UHoverComponent();
19
20     //Funcion para poder setear el hover con una tecla
21     void InputHover(UInputComponent* PlayerInputComponent);
22
23     //Funcion para hacer true el hover
24     void SetHover();
25
26     //Funcion Hover
27     void Hover(float DeltaTime);
28
29     //Funcion para volver a la posicion de inicio
30     void ResetPosition();
31
32 protected:
33     // Called when the game starts
34     virtual void BeginPlay() override;
35
36 public:
37     // Called every frame
38     virtual void TickComponent(float DeltaTime, ELevelTick TickType,
39                               FActorComponentTickFunction* ThisTickFunction) override;
40 private:
41
42     //Toggle para saber si iniciar el hover o no
43     UPROPERTY(EditAnywhere)
44     bool TriggerHover = true;
45
46     //Toggle para saber si resetear el hover
47     UPROPERTY(EditAnywhere)
48     bool Reset = true;
49
50     //Altura en la que queremos el dron
51     UPROPERTY(EditAnywhere)
52     float Height = 500.0f;
53
54     //Velocidad a la que queremos que se mueva
55     UPROPERTY(EditAnywhere)

```

2.2 Fase de prototipado

```
56     float Speed = 120.0f;
57
58     //Ubicaciones computadas o a computar
59     FVector StartRelativeLocation;
60     FVector HeightNorm; //Variable para
        normalizar el vector de la posición final y poder mover el dron de "
        poco en poco"
61     float MaxHeight = 0.0f;
62     float ActualHeight = 0.0f;
63
64 };
```

Listing 2.4: Código fuente del componente de escena HoverComponent

```
1 //Daniel Corrales 2023
2 //TFG UPM
3
4
5 #include "HoverComponent.h"
6
7
8 // Sets default values for this component's properties
9 UHoverComponent::UHoverComponent()
10 {
11     // Set this component to be initialized when the game starts, and to be
        ticked every frame. You can turn these features
12     // off to improve performance if you don't need them.
13     PrimaryComponentTick.bCanEverTick = true;
14
15     // ...
16 }
17
18 void UHoverComponent::InputHover(UInputComponent *PlayerInputComponent)
19 {
20     PlayerInputComponent->BindAction("Hover", IE_Pressed, this, &
        UHoverComponent::SetHover);
21 }
22
23 void UHoverComponent::SetHover()
24 {
25     TriggerHover = true;
26 }
27
28 void UHoverComponent::Hover(float DeltaTime)
29 {
30     //Establecemos la altura actual
31     ActualHeight += DeltaTime * Speed * 1;
32     //Computamos y actualizamos la ubicación en caso de no haber llegado a la
        altura máxima
33     if(ActualHeight <= MaxHeight || ActualHeight <= 0.0f){
34         SetRelativeLocation(StartRelativeLocation + HeightNorm *
        ActualHeight);
35     }
36 }
37
38 void UHoverComponent::ResetPosition()
39 {
40     FVector start;
41     start.X = 0;
42     start.Y = 0;
43     start.Z = 40.0f;
44     SetRelativeLocation(start);
45 }
46
47
48 // Called when the game starts
```

```

49 void UHoverComponent::BeginPlay()
50 {
51     Super::BeginPlay();
52
53     //Obtenemos la ubicacion inicial del dron
54     StartRelativeLocation = GetRelativeLocation();
55
56
57     //Computamos el movimiento normalizado y la ubicacion final del dron
58     HeightNorm = StartRelativeLocation;
59     HeightNorm.Z += Height;
60     HeightNorm.Normalize();
61     MaxHeight = Height;
62 }
63
64 // Called every frame
65 void UHoverComponent::TickComponent(float DeltaTime, ELevelTick TickType,
66     FActorComponentTickFunction* ThisTickFunction)
67 {
68     Super::TickComponent(DeltaTime, TickType, ThisTickFunction);
69
70     if(TriggerHover && !Reset)
71     {
72         Hover(DeltaTime);
73     }
74
75     if(Reset && !TriggerHover)
76     {
77         ResetPosition();
78     }
79 }

```

Tras desarrollar estos módulos de prueba basados en componentes de escena y observar su funcionamiento nuestro siguiente paso sería crear objetos basados en la unión de componentes independientes, lo que nos interesa en gran medida para proporcionarle cierta modularidad. La idea sería gestionar la comunicación entre los componentes desde un actor propio. Con este fin proponemos la creación de un peon e implementar en el mismo un control diferencial de la velocidad, prueba que describiremos más adelante.

2.2.2. Prueba de importación de mallas:

Durante el desarrollo de la prueba anteriormente descrita, busqué formas de poder integrar un modelo 3D, conocido como *mesh*, de un cuadricóptero en el editor de Unreal y como hacer que los componentes lo movieran.

Para empezar busqué en internet un modelo 3D gratuito de un cuadricóptero[referencia a la página del modelo 3D del dron]. Una vez descargado faltaba importarlo a la carpeta de contenidos de Unreal Engine para su uso. Cabe mencionar que estos modelos 3D han de ser exportados de sus respectivas aplicaciones en formato .fbx ya que este es el que reconoce Unreal. El proceso de importación es muy simple pudiendo simplemente arrastrar el archivo al editor y tras una simple ventana de configuración tener ya disponible el modelo para su uso. El modelo seleccionado fue el siguiente:

(foto del mesh del dron en el entorno)

2.2 Fase de prototipado

Si se desea Epic Games también cuenta con una plataforma en la que los usuarios pueden subir sus modelos 3D ya sea de niveles u objetos a una tienda online. La desventaja es que la mayoría del contenido es de pago.

Con esto quedaba solventado el problema de poder tener un modelo tridimensional de un dron. La siguiente cuestión será determinar si Unreal genera las colisiones automáticamente al proporcionarle una *hitbox* al modelo o si estas han de ser programadas desde 0.

2.2.3. Prueba de implementación de un control diferencial de la velocidad:

Finalmente y tras las primeras pruebas de movimiento decidimos crear un peón ya que este es el objeto ideal sobre el que crear un dron. Así pues, tras crear una clase en C++ de un peón, implementamos en la misma un control diferencial de la velocidad, un concepto simple basado en 3 vectores, uno para establecer la velocidad deseada, otro para definir la velocidad real del dron y un tercero que se encarga de calcular un delta con la velocidad real actual del dron, la velocidad objetivo y el tiempo transcurrido.

Es relevante considerar que unreal proporciona una función *tick* dentro de todas sus clases, la cual se ejecuta a cada frame generado por el motor y ejecuta las instrucciones descritas en la misma. Al mismo tiempo, esta función nos proporciona una variable *DeltaTime* la cual lleva la cuenta del tiempo que pasa dentro del editor durante la ejecución del programa. Esta función será la que más carga computacional tenga al final, dado que es la que utilizaremos para poder generar un movimiento fluido en un futuro.

Otra función generada automáticamente y que merece la pena considerar es *BeginPlay*, esta se ejecuta en el primer instante en que se entra al nivel creado y solo esa vez. Esta resulta útil para establecer variables iniciales o si por ejemplo se desea almacenar la posición inicial de algún objeto.

Una vez explicados los conceptos anteriores pasamos a la implementación del control en cuestión, una implementación simple y que nos permite configurar una velocidad máxima, una mínima y una constante para ajustar la velocidad en el cálculo del delta de la velocidad. La implementación queda tal que así:

Listing 2.5: Header del peón DronePawn

```
1 //Daniel Corrales 2023
2 //TFG UPM
3
4 #pragma once
5
6 #include "CoreMinimal.h"
7 #include "GameFramework/Pawn.h"
8 #include "DronePawn.generated.h"
9
10 UCLASS()
11 class DRONESIM_API ADronePawn : public APawn
12 {
13     GENERATED_BODY()
14 }
```

```

15 public:
16     // Sets default values for this pawn's properties
17     ADronePawn();
18
19     //Functions to modify the speed on each axis
20     void ModifySpeedX();
21     void ModifySpeedNegativeX();
22     void ModifySpeedY();
23     void ModifySpeedNegativeY();
24     void ModifySpeedZ();
25     void ModifySpeedNegativeZ();
26     void CalculateDelta();
27     void CalculateVelocity();
28
29 protected:
30     // Called when the game starts or when spawned
31     virtual void BeginPlay() override;
32
33 public:
34     // Called every frame
35     virtual void Tick(float DeltaTime) override;
36
37     // Called to bind functionality to input
38     virtual void SetupPlayerInputComponent(class UInputComponent*
39         PlayerInputComponent) override;
40 private:
41
42     //Variables for speed control
43     UPROPERTY(EditAnywhere)
44     FVector realVelocity;
45
46     UPROPERTY(EditAnywhere)
47     FVector deltaVelocity;
48
49     UPROPERTY(EditAnywhere)
50     FVector targetVelocity;
51
52     UPROPERTY(EditAnywhere)
53     float velocityConstant = 0.5f;
54
55     UPROPERTY(EditAnywhere)
56     float upperLimit = 1000.0f;
57
58     UPROPERTY(EditAnywhere)
59     float lowerLimit = -1000.0f;
60
61
62 };

```

Listing 2.6: Código fuente del peón DronePawn

```

1 //Daniel Corrales 2023
2 //TFG UPM
3
4 #include "DronePawn.h"
5
6 // Sets default values
7 ADronePawn::ADronePawn()
8 {
9     // Set this pawn to call Tick() every frame. You can turn this off to
10     // improve performance if you don't need it.
11     PrimaryActorTick.bCanEverTick = true;
12 }
13

```

2.2 Fase de prototipado

```
14 // Called every frame
15 void ADronePawn::Tick(float DeltaTime)
16 {
17     Super::Tick(DeltaTime);
18
19     realVelocity = GetActorLocation() * DeltaTime;
20     CalculateDelta();
21     CalculateVelocity(); //CalculateVelocity
22     FVector pos = GetActorLocation();
23     pos += DeltaTime*realVelocity;
24     SetActorLocation(pos);
25 }
26
27 // Called when the game starts or when spawned
28 void ADronePawn::BeginPlay()
29 {
30     Super::BeginPlay();
31
32     targetVelocity = {0.0f, 0.0f, 0.0f};
33 }
34
35 // Called to bind functionality to input
36 void ADronePawn::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
37 {
38     Super::SetupPlayerInputComponent(PlayerInputComponent);
39
40     //Setting up bindings to add or subtract from the targetVelocity
41     PlayerInputComponent->BindAction("XPositive", IE_Pressed, this, &ADronePawn
        ::ModifySpeedX);
42     PlayerInputComponent->BindAction("XNegative", IE_Pressed, this, &ADronePawn
        ::ModifySpeedNegativeX);
43     PlayerInputComponent->BindAction("YPositive", IE_Pressed, this, &ADronePawn
        ::ModifySpeedY);
44     PlayerInputComponent->BindAction("YNegative", IE_Pressed, this, &ADronePawn
        ::ModifySpeedNegativeY);
45     PlayerInputComponent->BindAction("ZPositive", IE_Pressed, this, &ADronePawn
        ::ModifySpeedZ);
46     PlayerInputComponent->BindAction("ZNegative", IE_Pressed, this, &ADronePawn
        ::ModifySpeedNegativeZ);
47 }
48
49
50 //Functions to set the speed with keyboard presses
51 void ADronePawn::ModifySpeedX()
52 {
53     targetVelocity.X += 10.0f;
54 }
55
56 void ADronePawn::ModifySpeedNegativeX()
57 {
58     targetVelocity.X -= 10.0f;
59 }
60
61 void ADronePawn::ModifySpeedY()
62 {
63     targetVelocity.Y += 10.0f;
64 }
65
66 void ADronePawn::ModifySpeedNegativeY()
67 {
68     targetVelocity.Y -= 10.0f;
69 }
70
71 void ADronePawn::ModifySpeedZ()
72 {
73     targetVelocity.Z += 10.0f;
74 }
```

```

75
76 void ADronePawn::ModifySpeedNegativeZ()
77 {
78     targetVelocity.Z -= 10.0f;
79 }
80
81 /*
82 Function to calculate the change needed in the velocity in order to
83 get the drone to the target velocity
84 */
85 void ADronePawn::CalculateDelta()
86 {
87     deltaVelocity = (targetVelocity - realVelocity) * velocityConstant;
88 }
89
90 /*
91 Function to add to the realVelocity the values needed
92 to make the drone reach the intended velocity
93 */
94 void ADronePawn::CalculateVelocity()
95 {
96     if(deltaVelocity.X >= upperLimit)
97     {
98         deltaVelocity.X = upperLimit;
99     }
100    if(deltaVelocity.X <= lowerLimit)
101    {
102        deltaVelocity.X = lowerLimit;
103    }
104    if(deltaVelocity.Y >= upperLimit)
105    {
106        deltaVelocity.Y = upperLimit;
107    }
108    if(deltaVelocity.Y <= lowerLimit)
109    {
110        deltaVelocity.Y = lowerLimit;
111    }
112    if(deltaVelocity.Z >= upperLimit)
113    {
114        deltaVelocity.Z = upperLimit;
115    }
116    if(deltaVelocity.Z <= lowerLimit)
117    {
118        deltaVelocity.Z = lowerLimit;
119    }
120
121    realVelocity += deltaVelocity;
122 }

```

Asimismo, gracias a utilizar un peón tenemos acceso a entradas de teclado mediante la función `SetupPlayerInputComponent`, la cual nos permite aumentar y disminuir los valores de las componentes del vector de la velocidad objetivo mediante teclas durante la ejecución del nivel. Muy relevante tener en cuenta que estos *keybinds* también han de configurarse dentro del editor de Unreal:

(insertar captura de la configuración del editor)

Y no solo eso, si no que además hemos de configurar el `DefaultPawn` del nivel al peón de la clase que hemos creado `DronePawn`, dado que es este “peón por defecto” el que recoge las entradas del teclado y las transfiere al propio peón. Este cambio solamene lo podemos realizar desde el código del modo de juego que se generó automáticamente al crear el proyecto en Unreal. Dentro del ar-

2.2 Fase de prototipado

chivo NombreProyectoGameModeBase.h y NombreProyectoGameModeBase.cpp hemos de añadir las 2 siguientes líneas (En mi caso el nombre de los archivos es droneSimGameModeBase.h y droneSimGameModeBase.cpp):

Listing 2.7: Header del modo de juego droneSimGameModeBase

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2 //Daniel Corrales 2023
3 //TFG UPM
4
5 #pragma once
6
7 #include "CoreMinimal.h"
8 #include "DronePawn.h"
9 #include "GameFramework/GameModeBase.h"
10 #include "droneSimGameModeBase.generated.h"
11
12 /**
13  *
14  */
15 UCLASS()
16 class DRONESIM_API ADroneSimGameModeBase : public AGameModeBase
17 {
18     GENERATED_BODY()
19
20     //Creamos un constructor para que nos cargue el dronePawn como el default
21     //pawn del juego
22 public:
23     ADroneSimGameModeBase();
24 };
```

Listing 2.8: Código fuente del modo de juego droneSimGameModeBase

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2 //Daniel Corrales 2023
3 //TFG UPM
4
5
6 #include "droneSimGameModeBase.h"
7 #include "DronePawn.h"
8
9 ADroneSimGameModeBase::ADroneSimGameModeBase()
10 {
11     //Instanciamos que el default pawn sea el dron
12     DefaultPawnClass = ADronePawn::StaticClass();
13 }
```

Una vez realizados todos estos cambios veremos que aparece esto en el editor al ejecutar y seleccionar el peón:

(imagen de los valores de los vectores)

Ya configurado y testado el correcto funcionamiento de las funciones previamente descritas, el siguiente paso es modificar la posición del propio peón con las velocidades calculadas.

Tras esta última prueba concluimos que el movimiento del dron basado en este control diferencial es lo necesario para simular correctamente el movimiento y poder realizar una integración con ROS 2 ya que pretendemos que el movimiento

se rija mediante la velocidad que se enviará mediante los servicios de mensajería ofrecidos por ese framework.

3. Title of next chapter

3.1. Overview

bla bla bla bla bla bla bla bla bla b la bla bla bla PPS bla bla bla bla bla bla
bla
bla
bla
bla
bla bla bla bla bla bla, see [Ric21, Mar63].

3.2. The next section

4. Title of next chapter

4.1. Overview

bla bla bla bla bla bla bla bla bla b la bla bla bla PPS bla bla bla bla bla bla
bla
bla
bla
bla
bla bla bla bla bla bla, see [Ric21, Mar63].

4.2. The next section

A. Title of the first appendix chapter

A.1. Overview

bla bla bla bla bla bla bla bla bla bla bla bla bla bla b la bla bla bla bla bla bla
bla
bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla roughness parameter
 R_a bla
bla bla bla bla bla bla bla bla bla bla, see [ISOa].

A.2. The next section

Bibliografía

- [A⁺06] P. Abgrall et al. A novel fabrication method of flexible and monolithic 3D microfluidic structures using lamination of SU-8 films. *J. Micromech. Microeng.*, 16: 113–121, 2006, <http://dx.doi.org/10.1088/0960-1317/16/1/016>.
- [ADK⁺03] Yu.S. Akishev, A.V. Dem'yanov, V.B. Karal'nik, A.E. Monich, and N.I. Trushkin. Comparison of the ac barrier corona with dc positive and negative coronas and barrier discharge. *Plasma Physics Reports*, 29: 90–100, 2003, <http://www.triniti.ru/CTF&VM/Articles/Demyanov3.pdf>.
- [AG07] P. Abgrall and A.-M. Gué. Lab-on-chip technologies: making a microfluidic network and coupling it into a complete microsystem—a review. *J. Micromech. Microeng.*, 17: R15–R49, 2007, <http://dx.doi.org/10.1088/0960-1317/17/5/R01>.
- [AST] ASTM e 1294-89 (1999), standard test method for pore size characteristics of membrane filters using automated liquid porosimeter, ASTM International. <http://www.astm.org/DATABASE.CART/WITHDRAWN/E1294.htm>.
- [AST05] S.N. Abolmasov, T. Shirafuji, and K. Tachibana. Submillimeter dielectric barrier discharges at atmospheric pressure: Edge effect. *IEEE Trans. Plasma Sci.*, 33: 941–948, 2005, <http://dx.doi.org/10.1109/TPS.2005.845093>.
- [B⁺01a] S.E. Babayan et al. Deposition of silicon dioxide films with a non-equilibrium atmospheric-pressure plasma jet. *Plasma Sources Sci. Technol.*, 10(4): 573–578, 2001, <http://dx.doi.org/10.1088/0963-0252/10/4/305>.
- [B⁺01b] N.A. Bullett et al. Direct imaging of plasma-polymerized chemical micropatterns. *Surf. Interface Anal.*, 31(11): 1074–1076, 2001, <http://dx.doi.org/10.1002/sia.1146>.
- [B⁺04] F. J. Blanco et al. Novel three-dimensional embedded SU-8 microchannels fabricated using a low temperature full wafer adhesive bonding. *J. Micromech. Microeng.*, 14: 1047–1056, 2004, <http://dx.doi.org/10.1088/0960-1317/14/7/027>.
- [B⁺07] R. Brandenburg et al. Antimicrobial treatment of heat sensitive materials by means of atmospheric pressure Rf-

- driven plasma jet. *Contrib. Plasma Phys.*, 47: 72–79, 2007, <http://dx.doi.org/10.1002/ctpp.200710011>.
- [BCG09] M.C. Breadmore, E. Candish, and R.M. Guijt. Dry film photoresist, saving time and money without compromising on performance. In *μTAS 2009*, 2009.
- [BNCK04] B. Bilenberg, T. Nielsen, B. Clausen, and A. Kristensen. Pmma to su-8 bonding for polymer based lab-on-a-chip systems with integrated optics. *J. Micromech. Microeng.*, 14: 814–818, 2004, <http://dx.doi.org/10.1088/0960-1317/14/6/008>.
- [BNGvdM02] A. Bogaerts, E. Neyts, R. Gijbels, and J. van der Mullen. Gas discharge plasmas and their applications. *Spectrochim. Acta. B Atom. Spectros.*, 57(4): 609–658, 2002, [http://dx.doi.org/10.1016/S0584-8547\(01\)00406-2](http://dx.doi.org/10.1016/S0584-8547(01)00406-2).
- [Boh89] Willi Bohl. *Technische Strömungslehre*. Vogel-Verlag, Würzburg, 8th edition, 1989.
- [Bru08] Henrik Bruus. *Theoretical Microfluidics*. Oxford University Press, 2008.
- [BSE06] K.H. Becker, K.H. Schoenbach, and J.G. Eden. Microplasmas and applications. *J. Phys. D: Appl. Phys.*, 39: R55–R70, 2006, <http://dx.doi.org/10.1088/0022-3727/39/3/R01>.
- [BWMK05] R. Brandenburg, H.-E. Wagner, A.M. Morozov, and K.V. Kozlov. Axial and radial development of microdischarges of barrier discharges in N₂/O₂ mixtures at atmospheric pressure. *J. Phys. D: Appl. Phys.*, 38(11): 1649–1657, 2005, <http://dx.doi.org/10.1088/0022-3727/38/11/003>.
- [C⁺04] J. Carlier et al. Integrated microfluidics based on multilayered su-8 for mass spectrometry analysis. *J. Micromech. Microeng.*, 14: 619–624, 2004, <http://dx.doi.org/10.1088/0960-1317/14/4/024>.
- [Car03] W.R. Carr. On the laws governing electric discharges in gases at low pressures. *Phil. Trans. Roy. Soc. Lond. Math. Phys. Sci.*, 201: 403–433, 1903, <http://dx.doi.org/10.1098/rsta.1903.0020>.
- [CBS67a] F.M. Charbonnier, C.J. Bennette, and L.W. Swanson. Electrical breakdown between metal electrodes in high vacuum. i. theory. *J. Appl. Phys.*, 38: 627–633, 1967, <http://dx.doi.org/10.1063/1.1709385>.
- [CBS67b] F.M. Charbonnier, C.J. Bennette, and L.W. Swanson. Electrical breakdown between metal electrodes in high vacuum. ii. experimental. *J. Appl. Phys.*, 38: 634–640, 1967, <http://dx.doi.org/10.1063/1.1709386>.
- [CGW77] DJ Carlsson, A. Garton, and DM Wiles. The weatherability of

- polypropylene monofilaments. effects of fiber production conditions. *Journal of Applied Polymer Science*, 21(11): 2963–2978, 1977, <http://dx.doi.org/10.1002/app.1977.070211112>.
- [CK00] H.-K. Chang and Y.-K. Kim. UV-LIGA process for high aspect ratio structure using stress barrier and C-shaped etch hole. *Sensors and Actuators A*, 84: 342–350, 2000, [http://dx.doi.org/10.1016/S0924-4247\(00\)00408-8](http://dx.doi.org/10.1016/S0924-4247(00)00408-8).
- [CS96] J.J. Coogan and A.D. Sappey. Distribution of OH within silent discharge plasma reactors. *IEEE Trans. Plasma Sci.*, 24(1): 91–92, 1996, <http://dx.doi.org/10.1109/27.491706>.
- [Cut93] P. Cutler. Theory of electron emission in high fields from atomically sharp emitters: Validity of the fowler-nordheim equation. *Progress in Surface Science*, 42(1-4): 169–185, 1993, [http://dx.doi.org/10.1016/0079-6816\(93\)90068-7](http://dx.doi.org/10.1016/0079-6816(93)90068-7).
- [D⁺04] A.J. Dikmans et al. A new compact disc format of high density array synthesis applied to peptide nucleic acids and in situ MALDI analysis. *Molecular diversity*, 8(3): 197–207, 2004, <http://dx.doi.org/10.1023/B:MODI.0000036234.66238.4e>.
- [Dav98] J.R. Davis. *Metals handbook: desk edition*. ASM International, 2nd edition, 1998.
- [DK03] R. Dorai and M.J. Kushner. A model for plasma modification of polypropylene using atmospheric pressure discharges. *J. Phys. D: Appl. Phys.*, 36(6): 666–685, 2003, <http://dx.doi.org/10.1088/0022-3727/36/6/309>.
- [Dor02] Rajesh Dorai. *Modeling of Atmospheric Pressure Plasma Processing of Gases and Surfaces*. PhD thesis, University of Illinois at Urbana-Champaign, 2002.
- [Dus23] S. Dushman. Electron emission from metals as a function of temperature. *Physical Review*, 21: 623–636, 1923, <http://dx.doi.org/10.1103/PhysRev.21.623>.
- [E⁺03] J.G. Eden et al. Microplasma devices fabricated in silicon, ceramic, and metal/polymer structures: arrays, emitters and photodetectors. *J. Phys. D: Appl. Phys.*, 36: 2869–2877, 2003, <http://dx.doi.org/10.1088/0022-3727/36/23/001>.
- [EK91] B. Eliasson and U. Kogelschatz. Modeling and applications of silent discharge plasmas. *IEEE Trans. Plasma Sci.*, 19(2): 309–323, 1991, <http://dx.doi.org/10.1109/27.106829>.
- [ETMK05] M. Eichler, M. Thomas, H. Mewes, and C.-P. Klages. Orts-selektive Atmosphärendruck-Plasmavorbehandlung für das Direkt-Wafer-Bonden bei niedrigen Temperaturen. In *Mikrosystemtechnik Kongress 2005 : 10. bis 12. Oktober 2005 in Freiburg*.

- Berlin: VDE-Verlag, 2005, pp. 115-118, 2005, <http://www.vde-verlag.de/data/buecher.php?action=bookdetail&vertriebsnr=562926>.
- [F⁺03] K.Saalmächter F.Kleinschmidt, U.Stöhr et al. Nmr spectroscopy characterization and self-diffusion study of partially swollen polymer networks. *Macromol. Chem. Phys.*, 204: F24–F25, 2003.
- [FCKS01] M. Fahland, C. Charton, V. Kirchhoff, and U. Stöhr. Deposition of multilayer optical coatings onto plastic webs by pulse magnetron sputtering; challenges and opportunities. In *Proceedings 6th. International Symposium on Sputtering and Plasma Processes (ISSP)*, p. 189, Kanazawa, Japan, 2001.
- [FN28] R.H. Fowler and L. Nordheim. Electron emission in intense electric fields. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 119(781): 173–181, May 1928, <http://www.jstor.org/pss/95023>.
- [FNC02] W.W. Flack, H.A. Nguyen, and E. Capsuto. Characterization of a novel photoresist redistribution material for advanced packaging applications. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4690, pages 1203–1216, 2002, <http://dx.doi.org/10.1117/12.474198>.
- [For99] R.G. Forbes. Refining the application of fowler-nordheim theory. *Ultramicroscopy*, 79: 11–23, 1999, [http://dx.doi.org/10.1016/S0304-3991\(99\)00097-2](http://dx.doi.org/10.1016/S0304-3991(99)00097-2).
- [G⁺] Yu. B. Golubovskii et al. Homogeneous barrier discharge. Internet: <http://optics.phys.spbu.ru/~golub/english/barrier.html>.
- [GEB⁺08] I. Graz, A. Ebner, S. Bauer, C. Romanin, and H. Gruber. Micropatterned atmospheric pressure discharge surface modification of fluorinated polymer films for mammalian cell adhesion and protein binding. *Appl. Phys. A*, 92: 547–555, 2008, <http://dx.doi.org/10.1007/s00339-008-4614-z>.
- [GEK] GEKA mbH. Destroying chemical warfare agents and contaminated military waste. Internet: <http://www.geka-munster.de/Seiten-engl/vernichtung/plasma-index.html> (accessed 06-09-2009).
- [Ger03] M. Gerbracht. Trocknung dicker Photoresistschichten mittels unterschiedlicher Trocknungsverfahren. Diplomarbeit, Fachhochschule Düsseldorf, 2003.
- [GMBB01] Yu.B. Golubovskii, V.A. Maiorov, J. Behnke, and J.F. Behnke. The role of emission from the dielectric surface in a homogeneous barrier discharge in nitrogen. In *Proc. of 15th Int. Symp. on Plasma Chemistry (Orléans, France, 10-13 July 2001) vol VIII* p 3239-3244, 2001, http://optics.phys.spbu.ru/~golub/article-s/DBD_ISPC15.pdf.

- [GMBB02] Yu. B. Golubovskii, V. A. Maiorov, J. Behnke, and J. F. Behnke. Some aspects of the modeling of an uniform barrier discharge in nitrogen. In *Proc. of 16th ESCAM-PIG/5th ICRP (Grenoble, France, 14-18 July 2002)*, Vol. 1, p. 233-234, 2002, http://optics.phys.spbu.ru/golub/article-s/DBD_N2_16ESCAMPIG.pdf.
- [GMBB03] Yu.B. Golubovskii, V.A. Maiorov, J. Behnke, and J.F. Behnke. Modelling of the homogeneous barrier discharge in helium at atmospheric pressure. *J. Phys. D: Appl. Phys.*, 36: 39–49, 2003, <http://dx.doi.org/10.1088/0022-3727/36/1/306>.
- [GMS98] A. Goossens, E.L. Maloney, and J Schoonman. Gas-phase synthesis of nanostructured anatase TiO₂. *Chemical Vapor Deposition*, 4(3): 109–114, 1998, [http://dx.doi.org/10.1002/\(SICI\)1521-3862\(199805\)04:03<109::AID-CVDE109>3.0.CO;2-U](http://dx.doi.org/10.1002/(SICI)1521-3862(199805)04:03<109::AID-CVDE109>3.0.CO;2-U).
- [Gol] Yu. B. Golubovskii. Internet: <http://optics.phys.spbu.ru/~golub/images/filament.jpg> (accessed 07-09-2009).
- [GU93] S.R. Gaboury and M.W. Urban. Microwave plasma reactions of solid monomers with silicone elastomer surfaces: a spectroscopic study. *Langmuir*, 9(11): 3225–3233, 1993, <http://dx.doi.org/10.1021/la00035a076>.
- [GvJBB02] Yu.B. Golubovskii, V.A. Maiorov v J. Behnke, and J.F. Behnke. Stability of the barrier discharge relative to the filamentation. In *Proc. of 16th ESCAMPIG/5th ICRP (Grenoble, France, 14-18 July 2002)*, Vol. 1, p. 235-236, 2002, http://optics.phys.spbu.ru/golub/articles/barrier2d_16ESCAMPIG.pdf.
- [H⁺07] K. Hensel et al. Electrical and optical properties of AC microdischarges in porous ceramics. *Plasma Process. Polym.*, 4: 682–693, 2007, <http://dx.doi.org/10.1002/ppap.200700022>.
- [HBMB00] K. Handique, D.T. Burke, C.H. Mastrangelo, and M.A. Burns. Nanoliter liquid metering in microchannels using hydrophobic patterns. *Anal. Chem.*, 72(17): 4100–4109, 2000, <http://dx.doi.org/10.1021/ac000064s>.
- [HGB⁺98] M.O. Heuschkel, L. Guérin, B. Buisson, D. Bertrand, and P. Renaud. Buried microchannels in photopolymer for delivering of solutions to neurons in a network. *Sensors and Actuators B*, 48: 356–361, 1998, [http://dx.doi.org/10.1016/S0925-4005\(98\)00071-9](http://dx.doi.org/10.1016/S0925-4005(98)00071-9).
- [HLLL03] M. Han, W. Lee, S.-K. Lee, and S.S. Lee. 3D microfabrication with inclined/rotated UV lithography. *Sensors and Actuators A*, 111: 14–20, 2003, <http://dx.doi.org/10.1016/j.sna.2003.10.006>.
- [HNF⁺04] M. Heise, W. Neff, O. Franken, P. Muranyi, and J. Wunderlich. Sterilization of polymer foils with dielectric barrier discharge.

- ges at atmospheric pressure. *Plasmas polym.*, 9: 23–33, 2004, <http://dx.doi.org/10.1023/B:PAPO.0000039814.70172.c0>.
- [HNRA04] K.H. Ho, S.T. Newman, S. Rahimifard, and R.D. Allen. State of the art in wire electrical discharge machining (WEDM). *Int. J. Mach. Tool. Manufact.*, 44(12–13): 1247–1259, 2004, <http://dx.doi.org/10.1016/j.ijmachtools.2004.04.017>.
- [HSG03] M. Heule, U.P. Schönholzer, and L.J. Gauckler. Patterning colloidal suspensions by selective wetting of microcontact-printed surfaces. *J. Eur. Ceram. Soc.*, 24: 2733–2739, 2003, <http://dx.doi.org/10.1016/j.jeurceramsoc.2003.09.011>.
- [I⁺07] T. Ito et al. Characterization of a microfluidic device fabricated using a photosensitive sheet. *J. Micromech. Microeng.*, 17: 432–438, 2007, <http://dx.doi.org/10.1088/0960-1317/17/3/003>.
- [ISOa] ISO 10780:1994, stationary source emissions – measurement of velocity and volume flowrate of gas streams in ducts, ISO. http://www.iso.org/iso/catalogue_detail.htm?csnumber=18855.
- [ISOb] ISO 4003:1977, permeable sintered metal materials – determination of bubble test pore size, ISO. http://www.iso.org/iso/catalogue_detail.htm?csnumber=9678.
- [ISOc] ISO 4287 (1997), geometrical product specifications surface texture profile method terms, definitions and surface texture parameters, ISO. http://www.iso.org/iso/catalogue_detail.htm?csnumber=10132.
- [ITM98] N. Inagaki, S. Tasaka, and M. Makino. Plasma polymer deposition from mixture of tetramethoxysilane and oxygen on PET films and their oxygen gas barrier properties. *J. Appl. Polym. Sci.*, 64(6): 1031–1039, 1998, <http://dx.doi.org/10.1002/%28SICI%291097-4628%2819970509%2964%3A6%3C1031%3A%3AAID-APP1%3E3.0.CO%3B2-M>.
- [Jen02] K.L. Jensen. Theory of field emission. *Vacuum Microelectronics*, Chapter 3: 33–104, 2002, <http://dx.doi.org/10.1002/0471224332.ch3>.
- [Jen03] K.L. Jensen. Electron emission theory and its application: Fowler–Nordheim equation and beyond. *J. Vac. Sci. Technol. B*, 21(4): 1528–1544, 2003, <http://dx.doi.org/10.1116/1.1573664>.
- [JFG⁺01] R.J. Jackman, T.M. Floyd, R. Ghodssi, M.A. Schmidt, and K.F. Jensen. Microfluidic systems with on-line uv detection fabricated in photodefinable epoxy. *J. Micromech. Microeng.*, 11: 1–8, 2001, <http://dx.doi.org/10.1088/0960-1317/11/3/316>.
- [JJM⁺06] S. Jigueta, M. Judelewicz, S. Mischlerb, A. Bertchc, and P. Renaudc. Effect of filler behavior on nanocomposite SU8 photoresist

- for moving micro-parts. *Microelectron. Eng.*, 83: 1273–1276, 2006, <http://dx.doi.org/10.1016/j.mee.2006.01.068>.
- [JPB06] N. Jidenko, M. Petit, and J.P. Borra. Electrical characterization of microdischarges produced by dielectric barrier discharge in dry air at atmospheric pressure. *J. Phys. D: Appl. Phys.*, 39(2): 281–293, 2006, <http://dx.doi.org/10.1088/0022-3727/39/2/008>.
- [K⁺03] M.G. Kwak et al. Improvement of luminance efficiency in xenon dielectric barrier discharge flat lamp. *IEEE Trans. Plasma Sci.*, 31: 176–178, 2003, <http://dx.doi.org/10.1109/TPS.2003.808867>.
- [K⁺04] K. Kim et al. A tapered hollow metallic microneedle array using backside exposure of SU-8. *J. Micromech. Microeng.*, 14: 597–603, 2004, <http://dx.doi.org/10.1088/0960-1317/14/4/021>.
- [K⁺05] K.V. Kozlov et al. Barrier discharges in N₂/O₂ mixtures. *J. Phys. D: Appl. Phys.*, 38: 518–529, 2005, <http://dx.doi.org/10.1088/0022-3727/38/4/003>.
- [K⁺07] C.P. Klages et al. Surface technology with cold microplasmas. *Plasma Process. Polym.*, 4(3): 208–218, 2007, <http://dx.doi.org/10.1002/ppap.200600116>.
- [KBET07] C.P. Klages, C. Berger, M. Eichler, and M. Thomas. Microplasma-based treatment of inner surfaces in microfluidic devices. *Contributions to Plasma Physics*, 47: 49–56, 2007, <http://dx.doi.org/10.1002/ctpp.200710008>.
- [KE05] S.-O. Kim and J.G. Eden. Arrays of microplasma devices fabricated in photodefinable glass and excited AC or DC by interdigitated electrodes. *IEEE Photonics Technology Letters*, 17: 1543–1545, 2005, <http://dx.doi.org/10.1109/LPT.2005.848260>.
- [KFG⁺03] E. Kukhareuka, M.M. Farooqui, L. Grigore, M. Kraft, and N. Hollinshead. Electroplating moulds using dry film thick negative photoresist. *J. Micromech. Microeng.*, 13: S67–S74, 2003, <http://dx.doi.org/10.1088/0960-1317/13/4/311>.
- [KFK07] T. Koida, H. Fujiwara, and M. Kondo. Hydrogen-doped In₂O₃ as high-mobility transparent conductive oxide. *Jpn. J. Appl. Phys.*, 46(28): L685–L687, 2007, <http://dx.doi.org/10.1143/JJAP.46.L685>.
- [KHKT00] C.P. Klages, K. Höpfner, N. Kläke, and R. Thyen. Surface functionalization at atmospheric pressure by DBD-based pulsed plasma polymerization. *Plasmas and Polymers*, 5(2): 79–89, 2000, <http://dx.doi.org/10.1023/A:1009583815474>.
- [KJI⁺04] J. Kieninger, G. Jobst, G. Igel, I. Moser, and G. Urban. 3D polymer microstructures by laminating SU-8 films. In *Proceedings of μTAS 2004*, volume 2, pages 363–365, 2004.

- [KK08] Y.K. Kim and D. J. Kim. Deposition of super hydrophobic a-C:F films by dielectric barrier discharge at atmospheric pressure. In *Book of Abstract of 11th International Conference on Plasma Surface Engineering PSE 2008*, 2008, http://www.pse2008.net/pdf_edit/tmp/PO1063-PSE2008-2f1418.pdf.
- [Kog02] U. Kogelschatz. Filamentary, patterned, and diffuse barrier discharges. *IEEE Trans. Plasma Sci.*, 30(4): 1400–1408, 2002, <http://dx.doi.org/10.1109/TPS.2002.804201>.
- [Kog03] U. Kogelschatz. Dielectric-barrier discharges: Their history, discharge physics, and industrial applications. *Plasma Chem. Plasma Process.*, 23: 1–46, 2003, <http://dx.doi.org/10.1023/A:1022470901385>.
- [KPKH03] W.S. Kang, J.M. Park, Y. Kim, and S.H. Hong. Numerical study on influences of barrier arrangements on dielectric barrier discharge characteristics. *IEEE Trans. Plasma Sci.*, 31(4): 504–510, 2003, <http://dx.doi.org/10.1109/TPS.2003.815469>.
- [KPTK05] S. Kreitz, C. Penache, M. Thomas, and C.-P. Klages. Patterned DBD treatment for area-selective metallization of polymers-plasma printing. *Surf. Coat. Technol.*, 200: 676–679, 2005, <http://dx.doi.org/10.1016/j.surfcoat.2005.01.085>.
- [KTMI69] H. Kawauchi, K. Tuzimura, H. Maeda, and N. Ishida. Reaction of fluorescein-isothiocyanate with proteins and amino acids: II. preparation of fluorescein-thiohydantoin amino acids and their thin-layer chromatography. *Journal of Biochemistry*, 66(6): 783–789, 1969, <http://jlb.oxfordjournals.org/cgi/content/abstract/66/6/783>.
- [Kun00] E.E. Kunhardt. Generation of large-volume, atmospheric-pressure, nonequilibrium plasmas. *IEEE Trans. Plasma Sci.*, 28: 189–200, 2000, <http://dx.doi.org/10.1109/27.842901>.
- [LEKB08] N. Lucas, V. Ermel, M. Kurrat, and S. Büttgenbach. Microplasma stamps for selective surface modification: design and characterization. *J. Phys. D: Appl. Phys.*, 41: 215202 (7pp), 2008, <http://dx.doi.org/10.1088/0022-3727/41/21/215202>.
- [Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.*, 2(2): 164–168, 1944.
- [LHKB08] N. Lucas, A. Hinze, C.-P. Klages, and S. Büttgenbach. Design and optimization of dielectric barrier discharge microplasma stamps. *J. Phys. D: Appl. Phys.*, 41: 194012 (5pp), 2008, <http://dx.doi.org/10.1088/0022-3727/41/19/194012>.
- [Lid07] David R. Lide, editor. *CRC Handbook of Chemistry and Physics*. Taylor and Francis, 87th edition, 2007.

- [LOUY] L. Ledernez, F. Olcaytug, G. Urban, and H. Yasuda. Influence of the electrode material on the electrical breakdown of argon. accepted for Plasma Sources Science and Technology.
- [LPCE07] M. Lu, S.-J. Park, B.T. Cunningham, and J.G. Eden. Microcavity plasma devices and arrays fabricated by plastic-based replica molding. *J. Microelectromech. Syst.*, 16: 1397–1402, 2007, <http://dx.doi.org/10.1109/JMEMS.2006.886034>.
- [LPCE08] M. Lu, S.-J. Park, B.T. Cunningham, and J.G. Eden. Low temperature plasma channels generated in microcavity trenches with widths of 20–150 μm and aspect ratios as large as $10^4:1$. *Appl. Phys. Lett.*, 92: 1397–1402, 2008, <http://dx.doi.org/10.1063/1.2827197>.
- [LU05] B.A. Langowski and K.E. Uhrich. Microscale plasma-initiated patterning (μPIP). *Langmuir*, 21: 10509–10514, 2005, <http://dx.doi.org/10.1021/la052222m>.
- [Luc09] N. Lucas. *Microplasma Stamps - An Atmospheric-Pressure Plasma Source for the Area-Selective Modification of Surfaces*. PhD thesis, Institute for Microtechnology, TU Braunschweig, 2009.
- [M⁺00] S.F. Miralá et al. Electrical and optical diagnostics of dielectric barrier discharges (DBD) in He and N₂ for polymer treatment. *Plasmas and Polymers*, 5(2): 63–77, 2000, <http://dx.doi.org/10.1023/A:1009531831404>.
- [M⁺01] F. Massines et al. The role of dielectric barrier discharge atmosphere and physics on polypropylene surface treatment. *Plasmas and Polymers*, 6(1): 35–49, 2001, <http://dx.doi.org/10.1023/A:1011365306501>.
- [M⁺06] K. Misumi et al. Faster sensitivity and non-antimonite permanent photoresist for MEMS. In Qinghuang Lin, editor, *Advances in Resist Technology and Processing XXIII*, volume 6153 of *Proceedings of SPIE*, page 61534F. SPIE, 2006, <http://dx.doi.org/10.1117/12.656249>.
- [M⁺08] G.S. Malkov et al. Pulsed-plasma-induced micropatterning with alternating hydrophilic and hydrophobic surface chemistries. *Plasma Process. Polym.*, 5(2): 129–145, 2008, <http://dx.doi.org/10.1002/ppap.200700118>.
- [MA88] W.M. Mullins and B.L. Averbach. Surface properties of silicon and aluminum oxide powders. *Surface Science*, 206(1-2): 41–51, 1988, [http://dx.doi.org/10.1016/0039-6028\(88\)90013-1](http://dx.doi.org/10.1016/0039-6028(88)90013-1).
- [MA04] C.R. Martin and I.A. Aksay. Submicrometer-scale patterning of ceramic thin films. *J. of Electroceramics*, 12(1): 53–68, 2004, <http://dx.doi.org/10.1023/B:JECR.0000034001.15359.98>.

- [Mar63] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2): 431–441, 1963, <http://dx.doi.org/10.1137%2F0111030>.
- [MFR06] A. Mata, A.J. Fleischmann, and S. Roy. Fabrication of multi-layer SU-8 microstructures. *J. Micromech. Microeng.*, 16: 276–284, 2006, <http://dx.doi.org/10.1088/0960-1317/16/2/012>.
- [MFSM02] W. Menz, R. Förster, A. Schoth, and C. Müller. Non-conventional machining for microsystems. In *Proc. 3rd Euspen Int. Conf.*, page 37, 2002.
- [MG56] E.L. Murphy and R.H. Good, Jr. Thermionic emission, field emission, and the transition region. *Phys. Rev.*, 102: 1464 – 1473, 1956, <http://dx.doi.org/10.1103/PhysRev.102.1464>.
- [MPA⁺99] I. Müller, C. Punset, E. Ammelt, H.-G. Purwins, and J.P. Boeuf. Self-organized filaments in dielectric barrier glow discharges. *IEEE Trans. Plasma Sci.*, 27: 20–21, 1999, <http://dx.doi.org/10.1109/27.763004>.
- [MPGJ97] C. Martinet, V. Paillard, A. Gagnaire, and J. Joseph. Deposition of SiO₂ and TiO₂ thin films by plasma enhanced chemical vapor deposition for antireflection coating. *J. Non-Cryst. Solids*, 216: 77–82, 1997, [http://dx.doi.org/10.1016/S0022-3093\(97\)00175-0](http://dx.doi.org/10.1016/S0022-3093(97)00175-0).
- [MSTH96] K.A. Melzak, C.S. Sherwood, R.F.B. Turner, and C.A. Haynes. Driving forces for DNA adsorption to silica in perchlorate solutions. *J. Colloid Interface Sci.*, 181(2): 635–644, 1996, <http://dx.doi.org/10.1006/jcis.1996.0421>.
- [MSY⁺06] K. Misumi, K. Saito, A. Yamanouchi, T. Senzaki, and H. Honma. Minute tunnel structure formation with permanent film photoresist. *J. Photopolym. Sci. Technol.*, 19: 57–62, 2006, <http://dx.doi.org/10.2494/photopolymer.19.57>.
- [MT98] T. Masuzawa and T. Takawashi. Recent trends in EDM/ECM technologies in Japan. *VDI BERICHTE*, 1405: 1–16, 1998.
- [NDSL04] M. Noeske, J. Degenhardt, S. Strudthoff, and U. Lommatzsch. Plasma jet treatment of five polymers at atmospheric pressure: surface modifications and the relevance for adhesion. *International Journal of Adhesion & Adhesives*, 24: 171–177, 2004, <http://dx.doi.org/10.1016/j.ijadhadh.2003.09.006>.
- [NHKV09] D. Nußbaum, D. Herrmann, T. Knoll, and T. Velten. Micromixing structures for lab-on chip applications: Fabrication and simulation of 90° zigzag microchannels in dry film resist. In *4M/ICOMM 2009 Conference*, pages 215–218, 2009, <http://dx.doi.org/10.1243/17547164C0012009041>.
- [NHW01] A. Nakajima, K. Hashimoto, and T. Watanabe. Recent studies on super-hydrophobic films. *Monatsh*

- te für Chemie / Chemical Monthly*, 132: 31–41, 2001, <http://dx.doi.org/10.1007/s007060170142>.
- [Nie95] L. Niemeyer. A generalized approach to partial discharge modeling. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2(4): 510–528, 1995, <http://dx.doi.org/10.1109/94.407017>.
- [Nor28] L. Nordheim. Zur theorie der thermischen emission und der reflexion von elektronen an metallen. *Zeitschrift für Physik A Hadrons and Nuclei*, 46(11–12): 833–855, November 1928, <http://dx.doi.org/10.1007/BF01391020>.
- [O'B01] J. O'Brien. Advanced photoresist technologies for microsystems. *J. Micromech. Microeng.*, 11: 353–358, 2001, <http://dx.doi.org/10.1088/0960-1317/11/4/312>.
- [Ols98] Anders Olsson. *Valve-less Diffuser Micropumps*. PhD thesis, School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden, 1998.
- [OS99] A. Ohl and K. Schröder. Plasma-induced chemical micropatterning for cell culturing applications: a brief review. *Surf. Coat. Technol.*, 116: 820–830, 1999, [http://dx.doi.org/10.1016/S0257-8972\(99\)00150-4](http://dx.doi.org/10.1016/S0257-8972(99)00150-4).
- [Pas89] F. Paschen. Ueber die zum Funkenübergang in Luft, Wasserstoff und Kohlensäure bei verschiedenen Drucken erforderliche Potentialdifferenz. *Annalen der Physik*, 273(5): 69–96, 1889, <http://dx.doi.org/10.1002/andp.18892730505>.
- [PCI] PCI-SIG. PCI conventional specifications. Internet: <http://www.pcisig.com/specifications/conventional> (accessed 08-01-2010).
- [PGB⁺04] C. Penache, C. Gessner, T. Betker, V. Bartels, and C.-P. Klages. Plasma printing: patterned surface functionalisation and coating at atmospheric pressure. *IEE Proc.-Nanobiotechnol.*, 151: 139–144, 2004, <http://dx.doi.org/10.1049/ip-nbt:20040836>.
- [PHHS01] J. Park, I. Henins, H.W. Herrmann, and G.S. Selwyn. Discharge phenomena of an atmospheric pressure radio-frequency capacitive plasma source. *J. Appl. Phys.*, 89: 20–28, 2001, <http://dx.doi.org/10.1063/1.1323753>.
- [PJF02] W.G. Payne, B.W. James, and I.S. Falconer. The growth and decay of miniature dielectric barrier discharges in helium, neon, and argon. *IEEE Trans. Plasma Sci.*, 30: 170–171, 2002, <http://dx.doi.org/10.1109/TPS.2002.1003978>.
- [PKE05] S.-J. Park, K.S. Kim, and J.G. Eden. Nanoporous alumina as a dielectric for microcavity plasma devices: Multilayer Al/AlO structures. *Appl. Phys. Lett.*, 86: 221501, 2005, <http://dx.doi.org/10.1063/1.1923747>.

- [PP99] A.V. Phelps and Z.Lj. Petrovic. Cold-cathode discharges and breakdown in argon: surface and gas phase production of secondary electrons. *Plasma Sources Sci. Technol.*, 8: 21–44, 1999, <http://dx.doi.org/10.1088/0963-0252/8/3/201>.
- [PYS⁺02] C.-T. Pan, H. Yang, S.-C. Shen, M.-C. Chou, and H.-P. Chou. A low-temperature wafer bonding technique using patternable materials. *J. Micromech. Microeng.*, 12: 611–615, 2002, <http://dx.doi.org/10.1088/0960-1317/12/5/315>.
- [RBCW03] I. Radu, R. Bartnikas, G. Czeremuszkin, and M.R. Wertheimer. Diagnostics of dielectric barrier discharges in noble gases: Atmospheric pressure glow and pseudoglow discharges and spatio-temporal patterns. *IEEE Trans. Plasma Sci.*, 31(3): 411–421, 2003, <http://dx.doi.org/10.1109/TPS.2003.811647>.
- [RBW05] I. Radu, R. Bartnikas, and M.R. Wertheimer. Diagnostics and modelling of noble gas atmospheric pressure dielectric barrier discharges in homogeneous or diverging electric fields. *J. Phys. D: Appl. Phys.*, 38: 539–546, 2005, <http://dx.doi.org/10.1088/0022-3727/38/4/005>.
- [RCJ⁺07] A.D. Radadia, L. Cao, H.-K. Jeong, M.A. Shannon, and R.I. Masel. A 3D micromixer fabricated with dry film resist. In *Proc. of MEMS 2007*, pages 361–364, 2007, <http://dx.doi.org/10.1109/MEMSYS.2007.4433110>.
- [Ric21] O.W. Richardson. *The Emission of Electricity from Hot Bodies*. Longmans, Green and co., 1921.
- [RKML05] R.B. Rao, K.L. Krafcik, A.M. Morales, and J. Lewis. Microfabricated deposition nozzles for direct-write assembly of three-dimensional periodic structures. *Adv. Mater.*, 17(3): 289–293, 2005, <http://dx.doi.org/10.1002/adma.200400514>.
- [RRPR07] M. Radmilovic-Radjenovic, Z.L. Petrovic, and B. Radjenovic. Modelling of breakdown behavior by PIC/MCC code with improved secondary emission models. *J. Phys.: Conf. Ser.*, 71(1): 012007, 2007, <http://dx.doi.org/10.1088/1742-6596/71/1/012007>.
- [Rus21] Cristian Rus. Con 3.281 drones, este impresionante espectáculo en shanghai ha establecido un nuevo récord mundial. *xataka*, 2021, <https://www.xataka.com/drones/3-281-drones-este-impresionante-espectaculo-shanghai-ha-establecido-nuevo-record-mundial>.
- [S⁺00] R.C. Smith et al. Chemical vapour deposition of the oxides of titanium, zirconium and hafnium for use as high-k materials in microelectronic devices. a carbon-free precursor for the synthesis of hafnium dioxide. *Adv. Mater. Opt. and Electron.*, 10(3-5): 105–114, 2000, [http://dx.doi.org/10.1002/1099-0712\(200005/10\)10:3/5<105::AID-AMO402>3.0.CO;2-J](http://dx.doi.org/10.1002/1099-0712(200005/10)10:3/5<105::AID-AMO402>3.0.CO;2-J).

- [S⁺07a] A. Schenk et al. Microstructured electrode arrays: Optical analysis of the glow discharge in a magnified electrode gap. *Plasma Process. Polym.*, 4: S927–S930, 2007, <http://dx.doi.org/10.1002/ppap.200732301>.
- [S⁺07b] K. Stephan et al. Fast prototyping using a dry film photoresist: microfabrication of soft-lithography masters for microfluidic structures. *J. Micromech. Microeng.*, 17: N69–N74, 2007, <http://dx.doi.org/10.1088/0960-1317/17/10/N01>.
- [S⁺09] Uwe Stöhr et al. Multilayer photoresist stamps for selective plasma treatment in micrometer scales. *Plasma Process. Polym.*, 6(4): 228–233, 2009, <http://dx.doi.org/10.1002/ppap.200800217>.
- [S⁺10] U. Stöhr et al. Porous photoresist stamps for selective plasma treatment. *Plasma Processes Polym.*, 7(1): 9–15, 2010, <http://dx.doi.org/10.1002/ppap.200900063>.
- [SBL07] O. Stalter, B. Burger, and S. Lehrmann. Silicon Carbide (SiC) D-MOS for grid-feeding solar-inverters. In *European Conference on Power Electronics and Applications, 2007*, pages 1–10, 2007, <http://dx.doi.org/10.1109/EPE.2007.4417654>.
- [SBTML06] C. Sarra-Bournet, S. Turgeon, D. Mantovani, and G. Laroché. A study of atmospheric pressure plasma discharges for surface functionalization of PTFE used in biomedical applications. *J. Phys. D: Appl. Phys.*, 39(16): 3461–3469, 2006, <http://dx.doi.org/10.1088/0022-3727/39/16/S03>.
- [SC07] J.W. Suk and J. Cho. Capillary flow control using hydrophobic patterns. *J. Micromech. Microeng.*, 17(4): N11–N15, 2007, <http://dx.doi.org/10.1088/0960-1317/17/4/N01>.
- [SG99] U.P. Schönholzer and L.J. Gauckler. Ceramic parts patterned in the micrometer range. *Adv. Mater.*, 11(8): 630–632, 1999, http://www.cheric.org/research/tech/periodicals/doi.php?art_seq=646499.
- [SMS04] K.H. Schoenbach, M. Moselhy, and W. Shi. Self-organization in cathode boundary layer microdischarges. *Plasma Sources Sci. Technol.*, 13(1): 177–185, 2004, <http://dx.doi.org/10.1088/0963-0252/13/1/023>.
- [Soc94] G. Socrates. *Infrared and Raman characteristic group frequencies: tables and charts*. John Wiley & Sons, 2nd edition, 1994.
- [SOK95] Y. Sawada, S. Ogawa, and M. Kogoma. Synthesis of plasma-polymerized tetraethoxysilane and hexamethyldisiloxane films prepared by atmospheric pressure glow discharge. *J. Phys. D: Appl. Phys.*, 28: 1661–1661, 1995, <http://dx.doi.org/10.1088/0022-3727/28/8/015>.

- [Sou] J. R. Southern. Toronto thunderstorm. Internet: http://en.wikipedia.org/wiki/File:Krunckwerke_-_IMG_4515_%28by-sa%29.jpg (accessed 07-09-2009).
- [Sta06] S.M. Starikovskaia. Plasma assisted ignition and combustion. *J. Phys. D: Appl. Phys.*, 39(16): R265–R299, 2006, <http://dx.doi.org/10.1088/0022-3727/39/16/R01>.
- [SVH⁺08] U. Stöhr, P. Vulto, P. Hoppe, G. Urban, and H. Reinecke. High-resolution permanent photoresist laminate for microsystem applications. *J. Micro/Nanolith. MEMS MOEMS*, 7: 033009, 2008, <http://dx.doi.org/10.1117/1.2964217>.
- [TAI06] H.M.L. Tan, T. Akagi, and T. Ichiki. Localized plasma treatment of poly(dimethylsiloxane) surfaces and its application to controlled cell cultivation. *J. Photopolym. Sci. Technol.*, 19: 245–250, 2006, <http://dx.doi.org/10.2494/photopolymer.19.245>.
- [TEK06] M. Thomas, M. Eichler, and C.-P. Klages. Coating with atmospheric pressure plasma processes: From large area to μ -structures. In *49th Annual Technical Conference Proceedings*. Society of Vacuum Coaters, 2006, <http://tinyurl.com/Thomas06>.
- [The] The LyX Team. Lyx 1.6.x. Internet: <http://www.lyx.org>.
- [TJLH06] Y.-C. Tsai, H.-P. Jen, K.-W. Lin, and Y.-Z. Hsie. Fabrication of microfluidic devices using dry film photoresist for microchip capillary electrophoresis. *J. Chromatogr. A*, 1111: 267–271, 2006, <http://dx.doi.org/10.1016/j.chroma.2005.12.003>.
- [Tow10] J.S. Townsend. *The Theory of Ionization of Gases by Collision*. Constable, 1910.
- [Tow15] J.S. Townsend. *Electricity in Gases*. Clarendon Press, 1915.
- [TvHKB07] M. Thomas, M. von Hausen, C.-P. Klages, and P. Baumhof. Generation of stable coatings with carboxylic groups by copolymerization of MAA and VTMS using DBD at atmospheric pressure. *Plasma Process. Polym.*, 4: S475–S481, 2007, <http://dx.doi.org/10.1002/ppap.200731210>.
- [TWL03] D. Therriault, S.R. White, and J.A. Lewis. Chaotic mixing in three-dimensional microvascular networks fabricated by direct-write assembly. *Nature Materials*, 2(4): 265–271, 2003, <http://dx.doi.org/10.1038/nmat863>.
- [TYL⁺06] Y.-C. Tsai, S.-J. Yang, H.-T. Lee, H.-P. Jen, and Y.-Z. Hsieh. Fabrication of a flexible and disposable microreactor using a dry film photoresist. *J. Chin. Chem. Soc.*, 53: 683–688, 2006, <http://nr.stpi.org.tw/ejournal/ChiChemSociety/2006/EJ52-2006-683.pdf>.

- [U⁺72] S. Udenfriend et al. Fluorescamine: A reagent for assay of amino acids, peptides, proteins, and primary amines in the picomole range. *Science*, 178: 871–872, 1972, <http://dx.doi.org/10.1126/science.178.4063.871>.
- [V⁺05a] T. Velten et al. Packaging of bio-mems: Strategies, technologies, and applications. *IEEE trans. adv. packag.*, 28: 533–546, 2005, <http://dx.doi.org/10.1109/TADV.2005.858427>.
- [V⁺05b] P. Vulto et al. Microfluidic channel fabrication in dry film resist for production and prototyping of hybrid chips. *Lab on a Chip*, 5: 158–162, 2005, <http://dx.doi.org/10.1039/b411885e>.
- [V⁺06] P. Vulto et al. Selective sample recovery of dep-separated cells and particles by phaseguide-controlled laminar flow. *J. Micromech. Microeng.*, 16: 1847–1853, 2006, <http://dx.doi.org/10.1088/0960-1317/16/9/013>.
- [VIU07] P. Vulto, G. Igel, and G.A. Urban. Full wafer fabrication process for microfluidic glass chips with electroplated electrodes. In *Solid-State Sensors, Actuators and Microsystems Conference, 2007. TRANSDUCERS 2007.*, pages 117–120, 2007, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4300085.
- [VKIU06] P. Vulto, C. Klaunick, G. Igel, and G. Urban. tmRNA purification by electrophoretic filtration for genomic identification of bacteria on-a-chip. In *Proceedings of μ TAS 2006*, volume 2, pages 377–379, 2006.
- [VSH⁺05] K.D. Vora, B.Y. Shew, E.C. Harvey, J.P. Hayes, and A.G. Peele. Specification of mechanical support structures to prevent SU-8 stiction in high aspect ratio structures. *J. Micromech. Microeng.*, 15: 978–983, 2005, <http://dx.doi.org/10.1088/0960-1317/15/5/012>.
- [W⁺05] Y. Washio et al. Multilayer ultra thick resist development for MEMS. In *Proceedings of SPIE*, volume 5753, page 959, 2005, <http://dx.doi.org/10.1117/12.600227>.
- [WH06] C. Wang and X. He. Polypropylene surface modification model in atmospheric pressure dielectric barrier discharge. *Surface & Coatings Technology*, 201(6): 3377–3384, 2006, <http://dx.doi.org/10.1016/j.surfcoat.2006.07.205>.
- [WW04] J.D. Williams and W. Wang. Microfabrication of an electromagnetic power relay using SU-8 based UV-LIGA technology. *Microsystem Technologies*, 10: 699–705, 2004, <http://dx.doi.org/10.1007/s00542-004-0396-1>.
- [XLW⁺07] X. Xu, L. Li, S. Wang, L. Zhao, and T. Ye. Deposition of SiO_x films with a capacitively-coupled plasma at atmospheric

- pressure. *Plasma Sources Sci. Technol.*, 16(2): 372–376, 2007, <http://dx.doi.org/10.1088/0963-0252/16/2/021>.
- [Y⁺06] L. Yu et al. Adhesive bonding with su-8 at wafer level for microfluidic devices. *J. Phys.: Conf. Ser.*, 34: 776–781, 2006, <http://dx.doi.org/10.1088/1742-6596/34/1/128>.
- [Yas05] Hirotsugu Yasuda. *Luminous Chemical Vapor Deposition and Interface Engineering*, volume 122 of *Surfactant Science Series*. Marcel Dekker, 1st edition, 2005.
- [YHK96] K. Yamada, T. Haraguchi, and T. Kajiyama. Plasma-graft polymerization of vinyl monomers with reactive groups onto a surface of poly (p-phenylene terephthalamide) fiber. *J. Appl. Polym. Sci.*, 60(11): 1847–1853, 1996, [http://dx.doi.org/10.1002/\(SICI\)1097-4628\(19960613\)60:11<1847::AID-APP8>3.0.CO;2-L](http://dx.doi.org/10.1002/(SICI)1097-4628(19960613)60:11<1847::AID-APP8>3.0.CO;2-L).
- [ZCPC04] J. Zhang, M.B. Chan-Park, and S.R. Conner. Effect of exposure dose on the replication fidelity and profile of very high aspect ratio microchannels in SU-8. *Lab on a Chip*, 4: 646–653, 2004, <http://dx.doi.org/10.1039/b403304c>.
- [ZKN⁺01] X.P. Zou, E.T. Kang, K.G. Neoh, C.Q. Cui, and T.B. Lim. Surface modification of poly (tetrafluoroethylene) films by plasma polymerization of glycidyl methacrylate for adhesion enhancement with evaporated copper. *Polymer*, 42(15): 6409–6418, 2001, [http://dx.doi.org/10.1016/S0032-3861\(01\)00113-6](http://dx.doi.org/10.1016/S0032-3861(01)00113-6).
- [ZKUI95] J. Zhang., K. Kato, Y. Uyama, and Y. Ikada. Surface graft polymerization of glycidyl methacrylate onto polyethylene and the adhesion with epoxy resin. *J. Polymer Sci. Polymer Chem.*, 33(15): 2629–2638, 1995, <http://dx.doi.org/10.1002/pola.1995.080331509>.
- [ZU99] Y. Zhao and M.-W. Urban. Spectroscopic studies of microwave plasma reactions of maleic anhydride on poly (vinylidene fluoride) surfaces: crystallinity and surface reactions. *Langmuir*, 15(10): 3538–3544, 1999, <http://dx.doi.org/10.1021/la981310m>.

Nomenclatura

R_a	arithmetic average roughness
PPS	Polyphenylene sulfide
PPS	Polyphenylene sulfide